THREAT ANALYSIS

THREAT INTELLIGENCE

사이버 위협 추적 과정의 모든 것

사이버 위협 추적, 공격자의 단서를 포착하라

서로 다른 보안 사고 간 연관성을 논할 때, 많은 사람들이 공격의 배후를 먼저 궁금해 한다. 이 논제는 사람들의 흥미를 유발하고, 실제로 보안 연구가들이 많이 받는 질문이다. 하지만, 사이버 위협 연관성 추적은 단순히 배후를 밝히는 것을 넘어 유사한 사이버 위협에 대한 예방 및 대응에 기여한다. 이는 최근 주목 받고 있는 사이버 위협 인텔리전스(Cyber Threat Intelligence)와도 연관된다.

이번 글에서는 안전한 보안 환경 조성을 담당하는 보안 업체 관점에서 사이버 위협의 연관성을 추적하는 방법과 과정을 설명하고, 위협에 효과적으로 대응하고자 하는 보안 부서를 위한 제언 사항을 다룬다.

최근 사이버 위협의 특징

먼저, 최근 사이버 위협의 특징을 살펴보자. 공격자는 대체로 확실한 목적을 갖고 공격을 수행한다. 그 목적은 대부분 금전적 수익을 얻고 정보를 탈취하는 것이며, 최근에는 많은 사이버 범죄 조직들이 표적형 랜섬웨어로 이득을 취하고 있다.

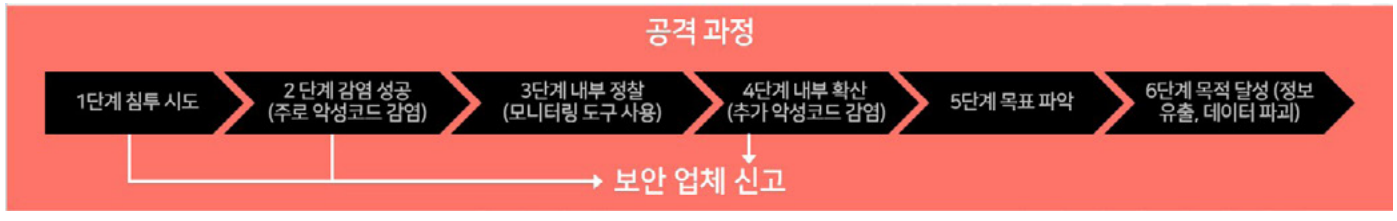
2020년 해외 사례들을 살펴보면, 대만중유공사(Chinese Petroleum Corporation: CPC), 일본 혼다(Honda), 미국 가민(Garmin), 일본 캡콤(Capcom) 등의 기업들이 표적형 랜섬웨어 피해를 입었다. 공격자는 내부 전산망에 침투해 정보를 유출하고 랜섬웨어를 배포했다. 그리고, 파일을 암호화해 업무를 마비시키고 유출된 민감한 정보를 공개하겠다며 돈을 요구했다. 자신의 존재를 노출하고 금전을 요구하는 것 외에는 기존 지능형 지속 위협(Advanced Persistent Threats: APT)와 형태가 유사하다. 일반적으로, APT 공격은 국가 차원의 지원을 받는 위협 그룹에서 수행한다고 여겨지지만, APT는 일종의 공격 방식이므로 어떤 공격자라도 수행이 가능하다.

국가적 지원을 받는 것으로 추정되는 위협 그룹(Threat Actor)은 보통 기밀 자료 유출을 목표로 한다. 전 세계를 상대로 활동하는 위협 그룹도 있지만, 주변국과 경쟁 혹은 갈등 관계에 관한 정보 수집이 우선인 경우가 많아 대체로 특정 지역 및 언어권에서 활동한다. 공격자 입장에서는 공격 대상의 낯선 언어와 문화가 공격에 장애 요인이 될 수 있지만, 반대로 해당 언어와 문화에 능숙하다면 이를 공격에 활용할 수 있다.

사이버 공격은 자동화되지 않고 상당 부분 사람이 수행한다. 인간의 특성상 익숙한 공격 방식을 선호하고 사용하는 악성코드나 도구도 유사한 경우가 많다. 그리고, 어이없는 실수로 자신의 흔적을 남기기도 한다. 방어자 입장에서는 공격자가 노리는 지역, 분야와 공격자의 공격 방식, 악성코드, 도구에 대한 정보를 알고 있으면 대응 방안을 마련할 수 있다.

공격의 전개 과정

악성코드를 이용한 공격은 보통 '침투 시도 ➜ 감염 성공 ➜ 내부 정찰 ➜ 내부 확산 ➜ 목표 파악 ➜ 목적 달성' 순으로 진행된다.



[그림 1] 일반적인 공격의 전개 과정

1단계: 침투 시도

침투 시도는 본격적인 공격 수행 전, 피싱 등을 통한 내부 사용자의 로그인 정보 수집 과정도 포함된다. 메일을 유포하거나 웹사이트 방문을 유도하는 악성코드 감염 시도가 가장 일반적이지만 기업이나 기관에서 사용하는 프로그램의 취약점을 통해 침입을 시도하기도 한다.

2단계: 감염 성공

이는 한번 혹은 여러 번의 침투 시도를 통해 악성코드 감염에 성공한 경우를 말한다. 감염된 시스템은 대부분 인터넷에 연결된 시스템이며, 공격자가 원하는 정보가 없는 경우가 많다. 공격자는 감염 시스템에 저장된 파일 등을 바탕으로 추가 모니터링이 필요하다고 판단되면 키로거(Keylogger), 녹화, 녹음 등의 추가 프로그램을 설치해 사용자를 감시한다.

3단계: 내부 정찰

내부 정찰 단계는 감염 시스템과 연결된 다른 시스템을 검색한다. 보통 포트 스캐너나 미미카츠(Mimikatz)와 같은 로그인 자격 증명(Credential) 정보 추출 프로그램을 이용해 추가 정보를 수집한다. 공격자는 공격 대상 조직에서 사용하는 각종 프로그램을 파악해 다음 단계의 공격 방안을 계획한다.

4단계: 내부 확산

내부 확산 단계는 파악된 내부 정보를 통해 연결된 시스템을 계속 장악해가는 과정이다. 원격 제어 프로그램을 사용하고 있다면 해당 프로그램의 로그인 정보를 이용해 다른 시스템에 접근할 수 있고 관리 목적 프로그램을 사용하는 경우, 해당 프로그램의 취약점을 찾거나 운영 서버 장악을 시도한다. 내부 시스템 장악 후, 목표 직원을 파악해 내부 업무 메일로 위장해 공격하면 악성코드 감염 성공 가능성이 더 높아진다. 새로운 시스템을 장악할 때마다 2단계에서 4단계 과정이 반복될 수 있다.

5단계: 목표 파악

목표 파악 단계는 공격자가 원하는 고객 정보, 내부 자료 등이 보관된 시스템을 확인하고 장악하는 과정이다. 공격자가 바로 목표한 시스템에 도달할 수도 있지만 외부에서 직접 연결할 수 없는 위치에 내부 자료가 저장된 시스템이 있으므로 1-4 단계를 반복해야 도달할 수 있다.

6단계: 목적 달성

목적 달성은 공격자가 원하는 최종 단계에 도달한 경우이다. 보통 공격자는 자료를 유출하고 유출 작업이 끝나면 자신의 흔적을 지운다. 하지만, 감시를 지속하거나 데이터를 손상하는 와이퍼(Wiper) 악성코드를 배포하기도 한다. 최근에는 데이터를 암호화하는 랜섬웨어(Ransomware)를 실행하는 경우도 있다.

공격당한 조직은 보통 1단계 침투 시도, 2단계 감염 성공, 4단계 내부 확산에서 발견되는 악성코드를 보안 업체에 신고한다. 다만, 보안 업체 입장에서 악성코드만으로는 공격 방식, 공격 성공 유무, 내부 피해를 완벽하게 알 수 없다. 관제 업체의 경우도 보통 외부 침투 시도 단계에서 방화벽, 침입 탐지, APT 방어 솔루션, 백신 프로그램에서 진단되는 내용을 중심으로 확인한다.

보안 업체에서 알 수 없는 내부 상황은 조직 내 보안 부서에서 파악해야 한다. 내부 시스템이 악성코드에 감염되는 단계에서 발견하면 다행이지만, 대다수의 경우 내부 정보 유출 확인 후 보안 업체나 관계 당국에 분석을 의뢰한다. 이 중, 몇 개월 심지어 1년 이상 침해 상태인 경우도 많다. 장기간 감염 상태라면 관련 로그가 존재하지 않아 조사를 하더라도 최초 유입 등의 정보를 파악하기 어렵다. 신속한 탐지와 대응이 무엇보다 중요한 이유다.

공격의 연관 관계 파악하기

위협 정보를 분석해 특정 위협 그룹들로 구분하기 위해서는 굉장히 다양한 내용들을 살펴봐야 한다. 단순히 유사한 악성코드가 발견됐다는 사실만으로 동일 그룹 소행이라 단정 짓기는 어렵다. 대신, 공격 방식, 공격 대상, 악성코드, 인프라, 언어, 실수 등을 종합적으로 판단해 연관 관계를 파악하고 그룹화 한다.

대분류	내용
공격 방식	공격 방식, 취약점
공격 대상	목표 국가, 언어, 분야 등
악성코드	기존 악성코드와 코드 유사 제작 언어, 컴파일러 버전, 리치 헤더(Rich Header), 빌드 시간 특징적 파일 경로, 파일 이름, 레지스트리, 뮤텍스(Mutex) 특징적 문자열 (제작자 별명, 암호, PDB 정보 등) Import, Export 함수 아이콘, 속성 (버전, 이름, 언어) 인증서 등
인프라	공격자 메일 주소, IP 특징적 URL C&C 서버 주소 C&C 서버의 웹쉘(Webshell) 및 운영 도구 등
언어	악성코드 내 포함 인프라 중 사용하는 언어 노출
실수	악성코드 내 제작자 흔적 실제 메일 주소 사용 실제 공격자 IP 어색한 문장 등

[표 1] 사이버 위협 그룹화 카테고리 및 내용

일반적으로 많은 위협 그룹이 사용하는 공격 방식이나 취약점은 위협 그룹의 특징으로 보기 어렵다. 하지만 특정 위협 그룹의 독특한 공격 방식은 특징이 될 수 있다. 예를 들어, 안다리엘(Andariel) 그룹은 한국산 액티브 X 프로그램과 관리 프로그램의 취약점을 이용한 공격을 즐겨 사용한다. 라자루스(Lazarus) 그룹은 다양한 공격 방식을 사용하지만, 오픈타입(OpenType) 폰트 취약점(CVE-2016-7256, MS16-132)을 이용한 공격 방식은 해당 그룹 외에는 공격 사례가 알려지지 않았다. 다만 공격 방식과 사용 취약점은 악성코드 샘플만으로는 확인하기 어려워, 심층적인 조사가 필요하다.

보안 업체는 고객의 악성코드나 피해 신고를 통해 공격자의 목적이 무엇인지 대략적으로 파악할 수 있다. 예를 들면, 국가 지원을 받는 위협 그룹의 경우 그룹 별로 공격 대상 및 동기가 확실한 편이다. 다만, 위협 그룹 마다 관심 분야가 다르므로 그룹별로 관심 있는 지역과 분야를 먼저 파악해야 한다. 그리고, 피해자가 고객이 아닐 경우에는 보안 업체도 공격 대상을 파악하는데 한계가 있어, 여러 보안 업체와 관계 당국의 정보를 취합해야 한다.

공격 분석 과정

공격자는 보통 악성코드와 도구를 이용해 침투 작업을 수행한다. 판매되는 원격 제어 도구를 사용하기도 하지만 보통 자체 제작한 악성코드를 활용한다. 공격자는 본인에게 익숙한 악성코드와 도구를 이용하는 경향이 있어, 이를 공격자의 특징으로 볼 수 있다. 다만, 악성코드의 종류가 주기적으로 바뀔 수 있고 동일 그룹이라고 해도 구성원에 따라 사용하는 악성코드와 도구가 조금씩 달라질 수 있다는 점을 염두에 둬야 한다.

분석가 입장에서, 악성코드의 유사성을 통한 연관 관계 파악은 쉽게 특징을 찾을 수 있어 가장 일반적으로 사용하는 방법이다. 하지만, 공격자가 악성코드를 여타 악성코드와 유사하게 만들어 다른 공격자로 위장할 수 있고, 생성기로 제작되거나 판매되는 악성코드를 활용할 경우 위 방법만으로 공격자를 특정할 수 없다. 또한, 자체 제작한 악성코드 중에도 공개된 소스코드를 가져와 재활용하는 경우도 있어, 단순히 악성코드의 코드 일부가 동일하다는 근거를 바탕으로 동일 그룹으로 분류하기는 어렵다.

악성코드 제작자는 새롭게 악성코드를 만들기도 하지만, 이미 완성된 악성코드 소스코드를 바탕으로 조금씩 수정해 나가기도 한다. 이 때, 조금 수정된 경우 전체적인 코드 흐름은 유사한 편이다. 2011년 3.4 디도스 공격에 사용된 악성코드는 특정 인터넷 공유 서비스를 통해 배포되었는데 2010년 9월 동일 업체에서 배포된 악성코드와 매우 유사하다.

```
if ( v2 == 0x22 )
{
  if ( *v1 != 0x22 )
    goto LABEL_6;
  while ( 1 )
  {
    ++v1;
LABEL_6:
    if ( !*v1 || *v1 > 0x20 )
    {
      StartupInfo.dwFlags = 0;
      GetStartupInfoA(&StartupInfo);
      GetProcessHeap_402A5F();
      sub_402A45(&dword_404000, (unsigned int)&unk_404004);
      if ( StartupInfo.dwFlags & 1 )
        v3 = StartupInfo.wShowWindow;
      else
        v3 = 10;
      v6 = v3;
      v4 = GetModuleHandleA(0);
      v5 = sub_402710(v4, 0, v1, v6);
      sub_402A77();
      ExitProcess(v5);
```

```
if ( v2 == 0x22 )
{
  if ( *v1 != 34 )
    goto LABEL_6;
  while ( 1 )
  {
    ++v1;
LABEL_6:
    if ( !*v1 || *v1 > 32 )
    {
      StartupInfo.dwFlags = 0;
      GetStartupInfoA(&StartupInfo);
      GetProcessHeap_402C0B();
      sub_402BF1(&unk_404000, &unk_404004);
      if ( StartupInfo.dwFlags & 1 )
        v3 = StartupInfo.wShowWindow;
      else
        v3 = 10;
      v6 = v3;
      v4 = GetModuleHandleA(0);
      v5 = sub_4028D0(v4, 0, v1, v6);
      sub_402C23();
      ExitProcess(v5);
```

[그림 2] 2010년 9월 배포 악성코드와 2011년 3월 배포 악성코드 비교

분석가는 코드의 유사성 외에도 다양한 특징들을 연관성 파악에 활용한다. 제작자가 악성코드를 제작할 때 사용하는 프로그래밍 언어와 컴파일러 버전도 제작자의 특징이 될 수 있다. 제작자마다 선호하고 익숙한 프로그래밍 언어가 있으며 일부는 독특한 언어를 사용하기도 한다. 예를 들어 틱 그룹(Tick Group)은 델파이(Delphi)로 제작된 악성코드가 많다.

파일 생성 시간과 운영 시간을 통해 공격자가 속한 시간대를 추정할 수도 있다. 공격자도 일을 하고 출퇴근을 하며 식사도 한다. 악성코드가 생성된 시간과 운영 서버의 활동 시간을 파악하면 공격자가 속한 시간대를 알 수 있다.

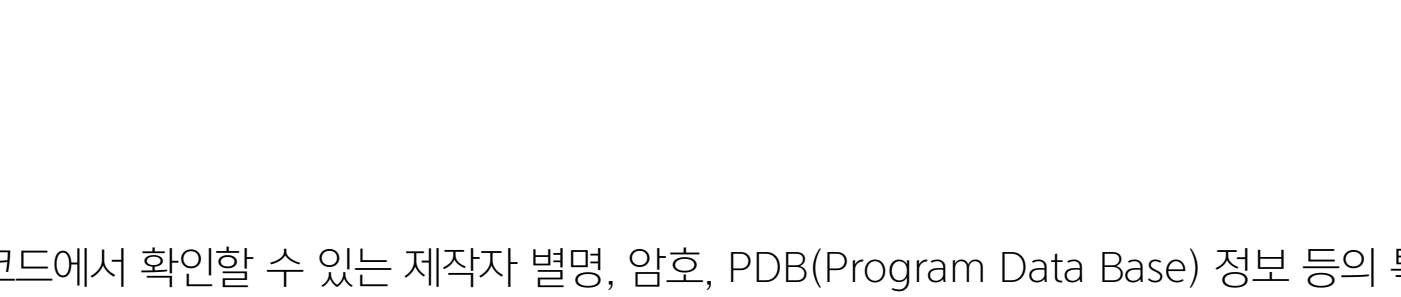
특징적인 파일 생성 경로나 이름도 유용한 정보가 된다. 공격자들도 매번 새로운 파일 이름을 사용하지 않고 동일하거나 유사한 파일 이름을 지속적으로 사용한다. 예를 들어, 마이킹즈(Mykings) 봇넷 제작자는 몇 년 동안 동일한 파일 이름을 사용하고 있다. 오션로터스 그룹은 정상 EXE 파일을 통해 악성 DLL 파일을 로딩한다. 종종 정상 EXE 파일을 변경하지만 동일한 파일을 몇 달 동안 사용해왔다. 공격자가 주로 사용하는 파일 이름을 알고 있으면 내부에서 동일한 파일 이름이 발견되었을 때 공격 시도를 신속하게 파악할 수 있다. 레지스트리 키나 뮤텍스(Mutex) 역시 동일하거나 비슷한 이름이 사용되는 경우가 흔히 있다.

악성코드에서 확인할 수 있는 제작자 별명, 암호, PDB(Program Data Base) 정보 등의 특징적 문자열도 공격자 추적에 도움이 된다.

[그림 3] 제작자 별명이 포함된 도구

이에 대한 구체적인 예시를 살펴보자.

아래 그림에서 의문의 'BM' 문자는 2009년 대한민국 정부 공격에 사용된 악성코드에서 처음 발견되었다.

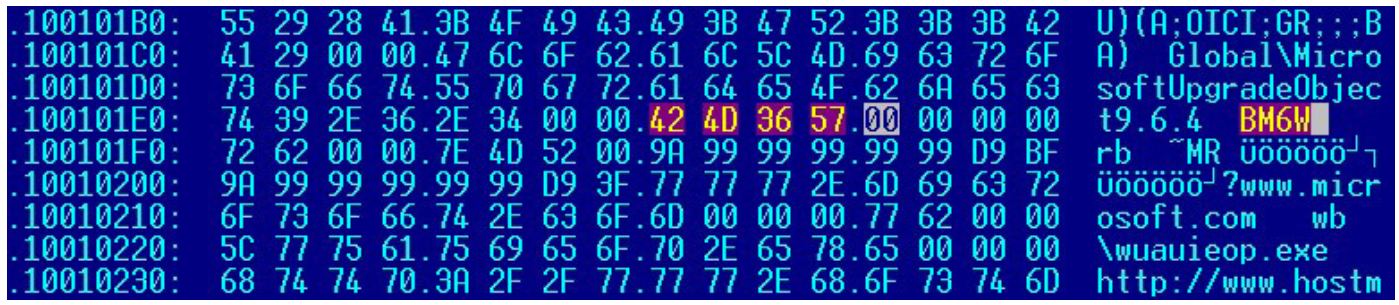


[그림 4] 2009년 발견된 악성코드 내 BM 문자열

다음은 2011년 발견된 라자루스 그룹 Redobot의 드롭퍼로, 'BMZA' 문자열을 포함하고 있다.

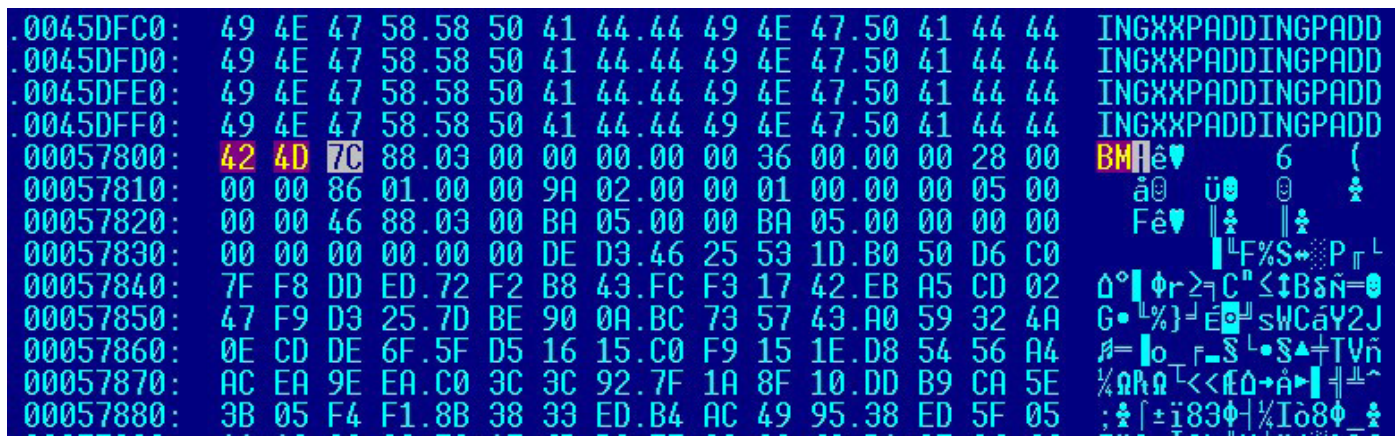
[그림 5] BMZA 문자열

2013년 6월 25일 6.25 사이버 공격 때 사용된 악성코드에는 'BM6W' 문자열이 존재한다.



[그림 6] 2013년 6.25 사이버 공격 악성코드 내 BM6W

2014년 Operation Black Mine에서 사용된 Bmdoor에서도 BM 문자열을 확인할 수 있다.



[그림 7] 2014년 Bmdoor 내 BM 문자열

'BM'이 무엇을 의미하는지 알 수 없지만 라자루스 그룹과 연관된 악성코드 제작자 중 누군가가 자신의 악성코드에 'BM' 문자를 의도적으로 넣어두고 있다. 이는 제작자 이니셜 가능성도 있다.

악성코드 제작자는 키로깅 정보 암호에 대해 동일한 암호 키를 장기간 사용하기도 한다. 2008년 대한민국 정부 기관 공격에 사용된 악성코드에서 발견된 암호화 키는 2012년 국내에서 발견된 악성코드에서도 사용되고 2019년 인도 원자력 발전소 공격에 사용된 악성코드에도 포함되어 있다. 단, 해당 암호키는 공개되어 있어 의도적으로 동일하게 사용했을 수 있으므로 악성코드 코드와 공격 방법 등 여러 측면에서 동일 그룹 소행인지 확인해야 한다.

일부 악성코드에는 PDB 경로가 포함되어 있고, 이를 통해 악성코드 제작자의 계정 이름, 악성코드 버전 등의 추가 정보를 파악할 수 있다. 2013년 3월 20일, 전산망 장애와 연관된 오퍼레이션 퍼스트 미션(Operation First Mission)에 사용된 악성코드는 2012년에 발견되었는데 다수의 변형이 PDB 경로를 포함하고 있었다.

PDB 내용
E:\Tong\Work\Op\1Mission\Team_Project\[2012.6 ~]\HTTP Trojan 2.0\HttpDr0pper\Win32\Release\HttpSecurityProvider.pdb
Z:\1Mission\Team_Project\[2012.1 ~ 3]\백도어\3Installer\Release\3Installer.pdb
Z:\1Mission\Team_Project\[2012.6 ~]\HttpDr0pper\Win32\Release\3PayloadDll.pdb
Z:\1Mission\Team_Project\[2012.6 ~]\HttpDr0pper\Win32\Release\HttpSecurityProvider.pdb
Z:\1Mission\Team_Project\[2012.6 ~]\HttpDr0pper\x64\Release\HttpSecurityProvider.pdb
Z:\1Mission\Team_Project\[2012.6 ~]\Keylogger\Win32\kylgr\Release\kylgr_mfc32.pdb
Z:\1Mission\Team_Project\[2012.6 ~]\Keylogger\x64\ConsoleKey\BIN\ConsoleKey.pdb
Z:\1Mission\Team_Project\[2012.6~]\HTTP Troy\HttpDr0pper\Win32\Release\HttpSecurityProvider.pdb
Z:\1Mission\Team_Project\[2012.6~]\HTTP Troy\HttpDr0pper\Win32\Release\3HttpDropper.pdb
Z:\1Mission\Tem_Project\[2012.6 ~]\HTTP Trojan 2.X\HttpDr0pper\x64\Release\HttpSecurityProvider.pdb

[표 2] Operation First Mission 악성코드의 PDB 내용

이 사례에서 PDB 정보를 통해 공격들에 대한 몇 가지 연관 정보를 확인할 수 있었다. 첫째, 공격은 1Mission, Team_Project 등 팀 단위로 운영 된다. 둘째, 일부 악성코드에는 한글로 '백도어'와 같은 문자열이 포함되어 있어 제작자는 한국어 사용자일 가능성이 높다. 셋째, 이 공격 작전은 2012년 1월부터 시작되었으며 2012년 6월부터 본격화되었다.

다른 변형은 'Z:\Make Troy\Concealment Troy\Exe_Concealment_Troy(Winlogon_Shell)\Dll\Concealment_Troy(Dll)\Release\Concealment_Troy.pdb' PDB 정보를 포함하고 있다. 공격자들은 악성코드 종류인 트로이목마(Trojan horse)를 Troy로 부르는 습관이 있

다. 3년 후 2015년 발견된 안다리엘 그룹의 Rifdoor 변형에서 'E:\Data\My Projfects\Troy Source Code\tcp1st\rifle\Release\rifle.pdb'와 같이 Troy 문자열을 사용한 악성코드가 발견됐으며, 둘의 연관 근거 중 하나로 보여진다.

[그림 8] Rifdoor의 PDB 경로

또 다른 예로, 툴라(Turla) 그룹의 스키퍼(Skipper) 악성코드 중 일부는 PDB 경로 문자열을 포함하고 있다.

일시	PDB 문자열
2017년 2월	C:\Users\work4\Documents\Visual Studio 2012\Projects\KOTEL 24.11.16 No COOKIE No STORAGE only BODY\KOTEL_2.1\Release\TerminateProcessTree.pdb
2017년 2월	C:\Users\work4\Documents\Visual Studio 2012\Projects\KOTEL 24.11.16 No COOKIE No STORAGE only BODY\KOTEL_2.1\x64\Release\GetPidByProcessName_x64.pdb'
2018 ~ 2019년	C:\Users\work4\Documents\Visual Studio 2012\Projects\Kotel without injecting\Release\dll_tranport.pdb
2019년 5월	C:\Users\George\Desktop\for B1 30.07.2018\Downloader without injecting\Downloader without injecting\Release\dll_tranport.pdb
2020년 3월	B:\PROJECTS\Kotel without injecting 15.11.2019\Release\dll_tranport.pdb
2020년 3월	B:\PROJECTS\Kotel without injecting 15.11.2019\Release\dll_terminate_process_tree.pdb

[표 3] 스키퍼 악성코드의 PDB 경로

이 문자열을 통해 제작자의 사용자 이름은 워크4(work4), 조지(George)로 유추할 수 있다. 또한, 악성코드 이름은 코텔(Kotel)이며 컴파일러 버전과 악성코드의 제작 시점도 추정 가능하다. 일/월/년도 표기 방식을 보면 유럽 지역 거주자로 예상된다. 제작자는 '트랜포트(tranport)'라는 단어를 계속 사용하는데 이는 '트랜스포트(transport)'의 오타로 보여 영어에 미숙할 가능성도 있다.

악성 DLL 파일의 경우 동일하거나 유사한 이름의 Export 함수를 사용한다. 2016년 툴라 그룹 변형 스키퍼 악성코드의 Export 함수는 CI, CS, _GetTempPathA, SHR, kp이며 2019년 발견된 변형의 Export 함수는 CI, CS, _GetTempPathA, SHR, SHRforCS로 2016년 버전과 유사하다.

```
1   .10001FD0 CI
2   .10001B80 CS
3   .100015C0 _GetTempPathA@8
4   .100018A0 SHR
5   .100020E0 kp
```

```
1   .10001DD0 CI
2   .10001B60 CS
3   .10001250 _GetTempPathA@8
4   .10001470 SHR
5   .100018B0 SHRforCS
```

[그림 9] 2016년 변형과 2019년 변형의 Export 함수

악성코드 제작자가 소스코드를 수정할 때 파일 정보의 아이콘과 속성은 변경하지 않는 경우가 있다. 이를 활용해 아이콘과 속성 정보를 바탕으로 유사한 악성코드를 찾을 수 있다. 또, 리소스 정보를 수정하지 않아 공격자가 사용하는 언어가 포함되기도 한다.

악성코드 제작자들은 인증서를 이용해 디지털 서명을 한다. 보안 프로그램에서 정상 인증서로 서명된 파일은 약간의 의심스러운 행위를 해도 허용하는 경우가 많기 때문이다. 따라서, 프로그램 제작사를 해킹해 실제 디지털 인증서 파일을 훔쳐 악성코드에 서명한다. 유출 인증서로 서명된 파일을 분석해 추가 악성코드를 발견하는 경우도 있다.

악성코드는 명령을 받기 위해 C&C 서버와 통신하는데 공격자가 C&C 서버를 직접 구축하거나 기존 서버를 해킹해 사용하기도 한다. 공격자에게도 매번 새로운 서버를 구축하는 것은 힘든 일이므로, 이전에 사용한 C&C 서버를 재활용하는 경우가 많고 같은 C&C 주소가 재활용됐다면 동일 그룹의 근거로 판단할 수 있다. 하지만, 다른 공격자가 동일 C&C 서버를 이용했을 가능성

도 배제해서는 안된다.

공격자가 C&C 서버와 이를 관리하는 서버를 별도로 둘 때도 있다. 수사하는 입장에서는 침해 사고를 조사하면서 공격자가 사용한 인프라를 분석하고, 공격자가 사용한 웹쉘(Webshell)이나 관리 프로그램의 특징을 바탕으로 동일 그룹으로 분류할 수 있다. 공격자는 인프라에 사용되는 프로그램에 대해서는 업데이트를 잘 하지 않고 과거 프로그램을 그대로 이용하기도 한다. 참고로, 공격자의 운영 서버는 법적인 문제 등으로 보통 국가 기관에서 주로 분석하며 보안 업체는 주로 C&C 서버와의 통신 체계를 분석한다.

공격자가 사용한 언어가 악성코드 내에 포함되기도 한다. 한글로 작성된 문서 파일에 특정 국가에서만 사용하는 폰트가 포함되는 경우가 있으며, 악성코드 내 잘못된 어법은 제작자가 해당 언어권 사용자가 아닐 수 있다는 근거가 된다. 일례로, 레드아이즈 그룹의 악성코드 드롭퍼는 '서버와의 련결이 실패하였습니다'라는 오류 메시지를 포함하며 한국에서 사용되는 '연결' 대신 사용하지 않는 '련결'을 사용했다.

```
00401180        push    ebp
00401181        mov     ebp, esp
00401183        sub     esp, 194h
00401189        mov     eax, ___security_cookie
0040118E        xor     eax, ebp
00401190        mov     [ebp+var_4], eax
00401193        push    ebx
00401194        push    500h
00401199        push    offset aI_0     ; "서버와의 련결이 실패하였습니다."
0040119E        push    offset aS       ; "%s "
004011A3        mov     ecx, 0C8h
004011A8        lea     ebx, [ebp+Text]
004011AE        call    sub_401080
004011B3        add     esp, 0Ch
004011B6        pop     ebx
004011B7        test    eax, eax
004011B9        js      short loc_4011D1
004011BB        push    10h             ; uType
004011BD        push    offset Caption  ; lpCaption
004011C2        lea     eax, [ebp+Text]
004011C8        push    eax             ; lpText
004011C9        push    0               ; hWnd
004011CB        call    ds:MessageBoxW
```

[그림 10] 어색한 한국어 메시지 포함

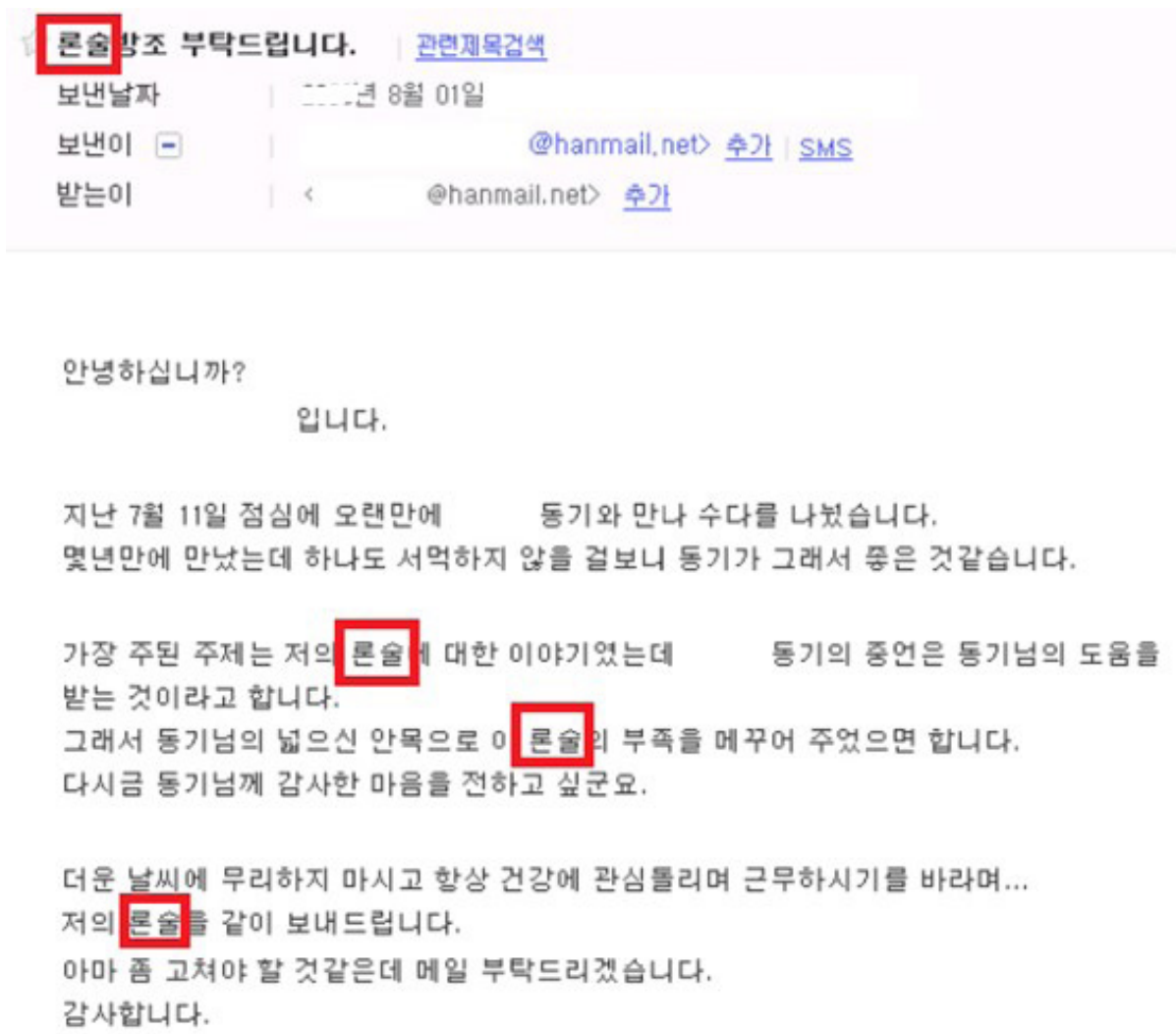
마지막으로, 공격자도 실수를 한다. 악성코드 내에 자신의 신분 정보를 남기거나 운영 서버 도메인 등록할 때 실제 메일 주소를 사용해 신분이 노출되기도 하고 VPN 실행을 깜빡하고 실제 IP를 노출하는 경우도 있다. 공격자가 예상하지 못하는 곳에서 공격자와 관련된 정보가 확인되기도 한다.

한 가지 연관 사례를 살펴보면, 레드아이즈 그룹의 악성 HWP 파일 내에서 'C:\Users\pad-2\AppData\Local\Temp\Hwp (3).exe', '\\192.168.100.22\saggazi\Happy\Work\2016.8~\2016.8.10~\(대학이름)\(한국인 이름)2016.8.24\2017.1.1메일\Hwp.exe' 경로 정보를 확인할 수 있었다. 참고로, HWP 문서에 파일을 삽입하면 전체 경로도 함께 저장된다. 이처럼 사용자의 실수를 통해 관련 정보를 얻기도 한다.

공격자 시스템은 pad-2와 싸가지(saggazi)가 있으며 싸가지 시스템은 한글로 대학 이름과 한국인 이름이 포함된 폴더에서 파일을 첨부 했음을 암시한다. 싸가지(saggazi)나 한국어 폴더 이름을 볼 때, 제작자는 한국인 혹은 한국어에 익숙한 사람으로 보이며 년도/월/일 표기 방식에서 아시아 지역 사람으로 보인다.

[그림 11] 악성 HWP에 포함된 파일 경로

가끔씩, 공격자가 어이없는 문법적 실수를 범하기도 한다. 국내의 경우, 공격에 사용된 메일 내용에서 한국에서는 사용하지 않는 표현이 보이곤 한다. 가장 흔하게 볼 수 있는 내용은 두음법칙 무시다. 이런 실수는 한국 정부에 대한 공격이 본격적으로 시작된 2008년부터 발견되었으며, 최근에도 종종 확인된다.



[그림 12] 어색한 한국어가 포함된 메일

이처럼 다양한 방법으로 공격 간 연관성을 파악할 수 있으며, 이 외에도 여러 사항들을 분석에 활용할 수 있다.

사이버 위협 추적 과정

보안 업체의 사이버 위협 추적은 '수집 ➜ 검토 ➜ 선정 ➜ 추적 ➜ 분석 ➜ 정리 ➜ 공유' 순서로 진행된다. 이는 필자의 개인적 의견이며 다른 순서로 진행될 수 있다.

1단계: 정보 수집

사이버 위협 추적의 첫걸음은 정보 수집이며, 이는 지속적으로 수행되어야 한다. 정보 수집은 뉴스, 소셜미디어, 분석 보고서 등 오신트(Open Source Intelligence: OSINT)부터 신고 센터 접수 샘플, 고객 문의, 보안 컨퍼런스, 위협 정보 공유 회의 등 다양한 경로를 통해 이뤄진다. 특히, 많은 보안 업체 연구가들이 소셜미디어 계정을 통해 새롭게 발견되는 보안 위협 정보를 공유한다.

2단계: 검토

다양한 경로로 수집된 사이버 위협 정보를 검토한다. 전 세계 모든 사이버 위협을 파악하는 것이 이상적인 시나리오지만 현실적인 문제로 범위를 제한할 수밖에 없다. 필자의 경우, 한국 기업과 정부를 노린 공격 및 지정학적으로 공격 가능성이 높은 동아시아권 위협 그룹들을 중점적으로 보는 편이다. 언론 보도의 경우, 보안 업체나 기관에서 공개한 원본 분석 내용을 확인해 침해 지표가 있는지 확인한다. 무작위 공격, 정보 확인이 어려운 내용 등 중요도가 낮은 사이버 위협을 제외하고, 중요도가 높은 위협 위주로 추가적인 정보가 있는지 파악한다.

3단계: 선정

다음으로, 분석할 위협 정보를 선정한다. 분석 대상을 선정할 때는 현재 활동 정도, 활동 기간, 활동 지역 등이 고려된다. 현재 한국이나 고객사를 왕성하게 공격하는 위협 그룹이 최우선이며 향후 고객사 공격 가능성이 높은 동아시아권 위협 그룹이 그 다음이다.

4단계: 추적

추적 과정을 통해 알려진 정보 외에 다른 정보가 있는지 확인한다. 추적은 관련 악성코드 혹은 그룹의 활동으로 추정되는 모든 내용을 수집하는 과정이다. 추적 과정에서 발견되는 내용이 많다면 우선 관련된 정보를 모두 수집 후, 실제 연관성이 있는지에 대한 분석은 별도로 진행한다. 연관성이 어느 정도 있다고 판단되지만 실제 분석해보면 별개의 사이버 위협일 수도 있다.

5단계: 분석

분석 과정은 추적을 통해 얻은 데이터를 확인해 실제 연관 관계가 존재하는지 확인하는 과정이다. 예를 들어 2020년 11월 모 기업에서 접수된 악성코드의 경우 추적 단계에서 2012년에 유사

하다고 생각되는 악성코드를 발견했지만 실제 악성코드 분석 결과, 파일 외형이 유사한 다른 악성코드였다.

추적과 분석 과정은 새로운 사실이 확인되면 반복적으로 진행될 수 있다. 이 과정에서 추가적인 정보가 확인되지 않거나 단기간 활동만 확인되면 관련 추적은 중단되기도 한다. 이후 추가 정보가 확인되면 다시 시작하는 경우도 있다. 일부 연구가들은 어떤 그룹이라고 답을 정하고 원하는 결론을 내리기도 한다. 상당히 위험한 접근 방식이며 사람들의 관심을 끌기 위해 무리하게 약한 연관성을 확대 해석하기도 한다. 따라서, 분석 과정 중 이미 공개된 분석 자료는 참고하되, 맹신해서는 안 된다.

6 & 7단계: 정리 및 공유

분석 과정을 통해 연관된 악성코드와 공격 사례를 정리하고 그 내용을 문서화해 내부 혹은 외부에 공유한다. 외부에 공개하기 어려운 내용은 내부나 인텔리전스 서비스를 통해 공개하고 그 외 자료는 대중에 공개된다.

사이버 위협 추적 실제 사례

다음으로, 안랩에서 작성한 일부 동향 보고서의 작성 과정을 통해 실제 사이버 위협 추적이 어떻게 이뤄지는지 살펴보고자 한다.

먼저, 일본 조총련 사이트를 통해 악성코드가 배포된 사건은 일본 보안업체 NTT 시큐리티의 보고서를 읽고 검토하는 것부터 시작했다. 단순히 조총련 홈페이지 해킹을 통한 악성코드 유포 사건으로 끝날 수 있었지만 추가 정보가 있다는 사실을 확인하고 추적을 통해 미국 내 북한 뉴스 사이트에 대한 추가 공격을 확인할 수 있었다. 안랩은 2019년 1월 관련 악성코드 분석 내용을 공개했으며 2020년 12월 15일 일본 경찰은 홈페이지를 해킹한 20대 한국인을 검찰에 송치했다(관련 보도). 관심 지역 & 분야의 분석 보고서가 큰 도움이 된 사례다.

2019년 4월 공개한 틱 그룹 동향 보고서는 공통 사안에 관심을 가진 사람과 인적 교류를 통해 정보를 얻을 수 있었다. 틱 그룹은 외국 보안 업체를 통해 한국과 일본에서 활동하고 있는 그룹이라 알려졌지만, 보고서에 공개된 샘플의 국내 활동은 확인되지 않은 상태였다. 필자가 일본 출

장에서 틱 그룹 분석 보고서를 쓴 일본 분석가와 일본 보안 컨퍼런스에서 알게 된 타이완 분석가를 통해 틱 그룹의 일본 내 추가 정보를 얻을 수 있었다. 다른 분석가들과 교류가 틱 그룹의 한국 활동 확인에 큰 도움이 되었다.

사이버 위협 추적 과정은 중간에 중단되기도 하고, 또 오랜 시간이 걸릴 수 있다. 일례로, 마이킹즈(Mykings) 봇넷은 2016년 11월 한국 회사의 인증서로 서명된 악성코드 발견으로부터 시작되었다. 2017년 초까지 관련 악성코드를 추적하다가 더 이상 추가 정보를 확인할 수 없어 추적을 중단했다.

그로부터 2년 뒤, 2019년 발생한 침해사고 조사 중 발견된 악성코드 파일 이름이 과거에 분석했던 마이킹즈 봇넷의 파일 이름과 동일해 비교한 뒤, 마이킹즈 봇넷으로 확인되어 재차 추적을 시작했다. 추가적인 추적과 분석을 통해 다른 보안 업체에서 이 악성코드를 마이킹즈 봇넷으로 명명했고, 여러 보안 회사에서 관련 분석 보고서를 공개했음을 알게 되었다. 안랩 역시 기존 보고서를 참고해 추가로 확인한 내용을 담아 분석 보고서를 공개했다.

언론에 공개된 보안 이슈를 확인하는 과정에서 새로운 위협 그룹을 발견하기도 한다. 일례로, 오퍼레이션 쉐도우 포스(Operation Shadow Force) 분석은 2020년 1월 일본 미쓰비시 전자 해킹 기사에서 시작됐다.

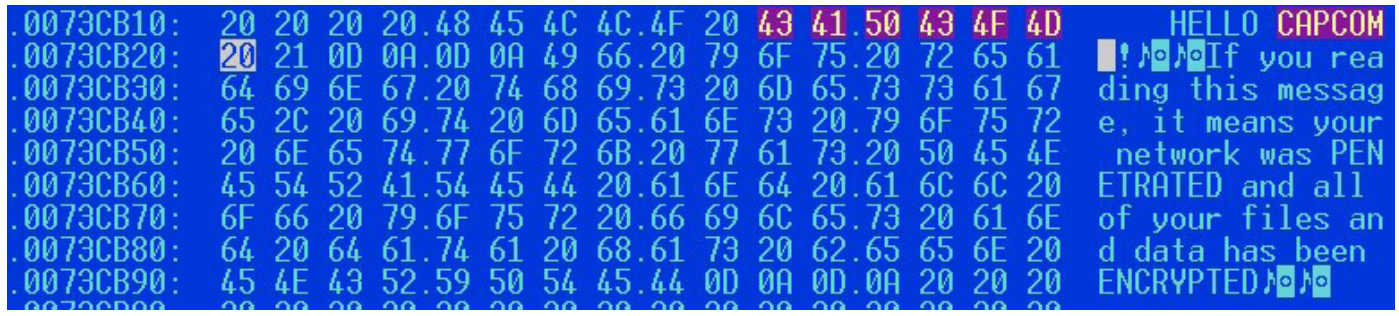
일본 언론보도에 따르면, 미쓰비시 해킹에 4개 그룹이 연관되어 있고 최초에는 틱 그룹으로 알려졌지만 현재는 블랙테크(BlackTech) 그룹을 배후로 추정하고 있다고 했다. 4개 그룹 중 엠디비(Emdivi)와 틱 그룹은 이미 분석된 내용이 있어 오로라 판다(Aurora Panda)와 블랙테크에 대한 분석을 시작했다.

오로라 판다는 외국에서 활동이 활발해 관련 보고서가 여럿 공개되어 있었다. 공개된 IOC를 바탕으로 관련 악성코드의 한국내 활동을 조사했지만 확인되지 않았다. 하지만, 관련 악성코드 중 한국 업체의 인증서로 서명되었다는 사실을 확인하고 해당 업체의 인증서로 서명된 파일을 분석하면서 새로운 악성코드를 발견했다. 추가적으로, 악성코드, 인증서, 피해 시스템에서 수집된 샘플 등을 통해 서로 연관성을 확인했다. 관련 내용은 2020년 4월에 Operation Shadow

Force로 공개했다.

블랙테크 그룹의 경우, 일본과 타이완을 주로 공격 대상으로 삼고 한국 기업과 유사한 C&C 서버도 발견되어 한국에서도 활동할 가능성이 있다고 예상했으나, 정황만 존재하고 최근 활동은 확인되지 않았다.

2020년 11월 4일에는 일본 게임 회사 캡콤이 라그나로커 랜섬웨어(Ragnar Locker Ransomware)에 감염되었다는 보도가 있었지만 관련 악성코드 정보는 확인되지 않았다. 이후, 2020년 11월 11일 일본 보안업체를 통해 의심 샘플이 공유됐다. 실제 캡콤을 노린 악성코드인지 확신할 수 없지만 일본에서 바이러스 토털에 파일을 업로드 했고, 악성코드 내부에 'CAPCOM' 문자열이 존재했다. 이는 기사에서 언급된 악성코드의 특징과 동일해 관련 악성코드일 가능성이 높을 것으로 판단했다. 또한, 관련 악성코드와 유사한 악성코드를 분석해 Birch, EDP, Omniga, Vg Cargo, Weglarzco 등의 업체가 피해를 입었다고 추정할 수 있었다.



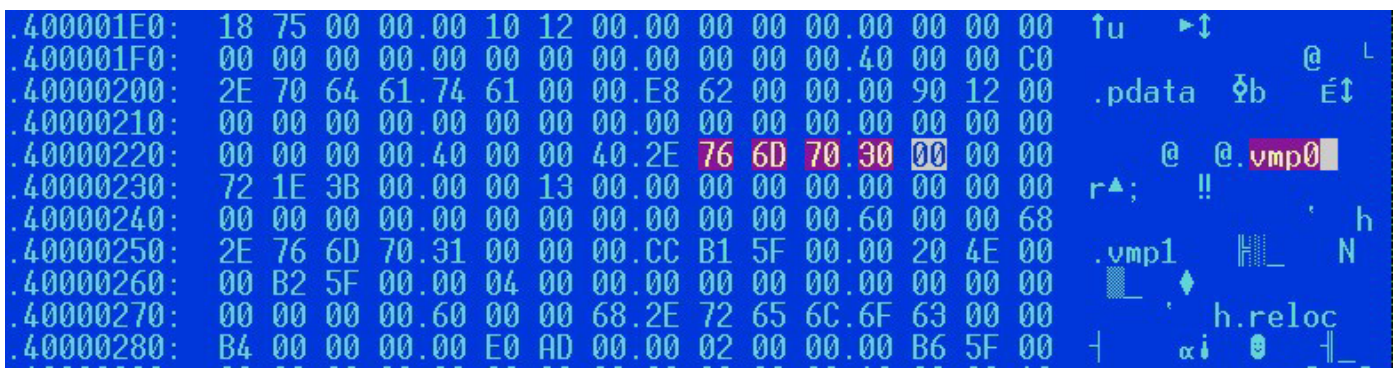
[그림 13] 랜섬웨어에서 확인되는 피해 업체

고객의 접수 샘플은 실제 활동하고 있는 악성코드를 파악할 수 있어 사이버 위협 추적의 단서를 제공한다. 오션로터스(OceanLotus) 그룹은 많은 분석가들이 베트남의 이익을 위해 활동하고 있다고 추정해왔다. 2019년 국내 기업으로부터 접수된 샘플을 오션로터스로 의심했지만 구체적인 근거는 확인되지 않았다. 이후, 2020년 본격적으로 오션로터스에 대한 분석을 시작하고 과거 오션로터스 악성코드와 국내에서 발견된 의심 악성코드를 비교해 오션로터스로 확인하고 관련 내용을 공개한 바 있다.

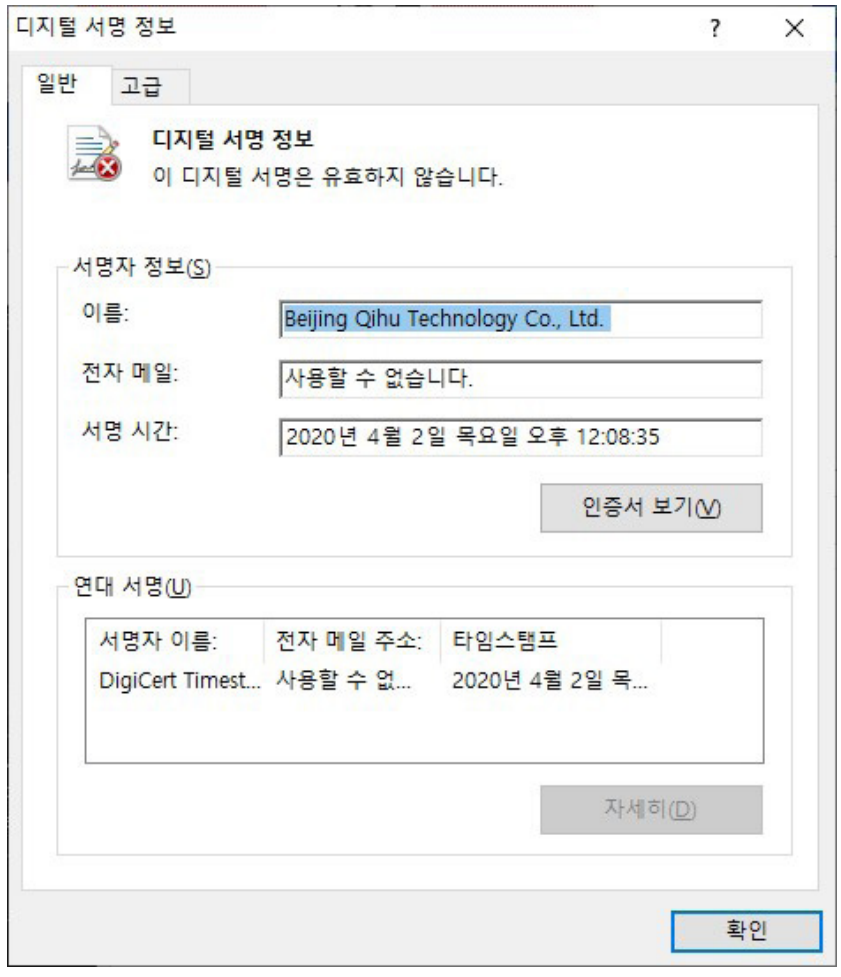
2020년 8월에는 VMProtect로 패킹된 미미카츠(Mimikatz) 샘플이 고객으로부터 접수되었

다. 미미카츠는 현재 공격자가 가장 널리 사용하고 있는 크리덴셜 정보를 훔치는 도구이지만 VMProtect로 패킹되어있고 주로 중국 업체의 인증서로 디지털 서명된 것처럼 위장하고 있다.

2020년 8월에는 VMProtect로 패킹된 미미카츠 샘플이 고객으로부터 접수되었다. 미미카츠는 현재 공격자가 가장 널리 사용하고 있는 크리덴셜 정보를 훔치는 도구이지만 VMProtect로 패킹되어있고 주로 중국 업체의 인증서로 디지털 서명된 것처럼 위장하고 있다.



[그림 14] VMProtect로 패킹된 미미카츠



[그림 15] 가짜 디지털 인증서로 서명

VMProtect 패킹과 가짜 인증서 서명, 그리고 유사한 파일 이름을 통해 공격자는 2020년 초부터 현재까지 꾸준히 활동하고 있다는 점을 파악할 수 있었다.

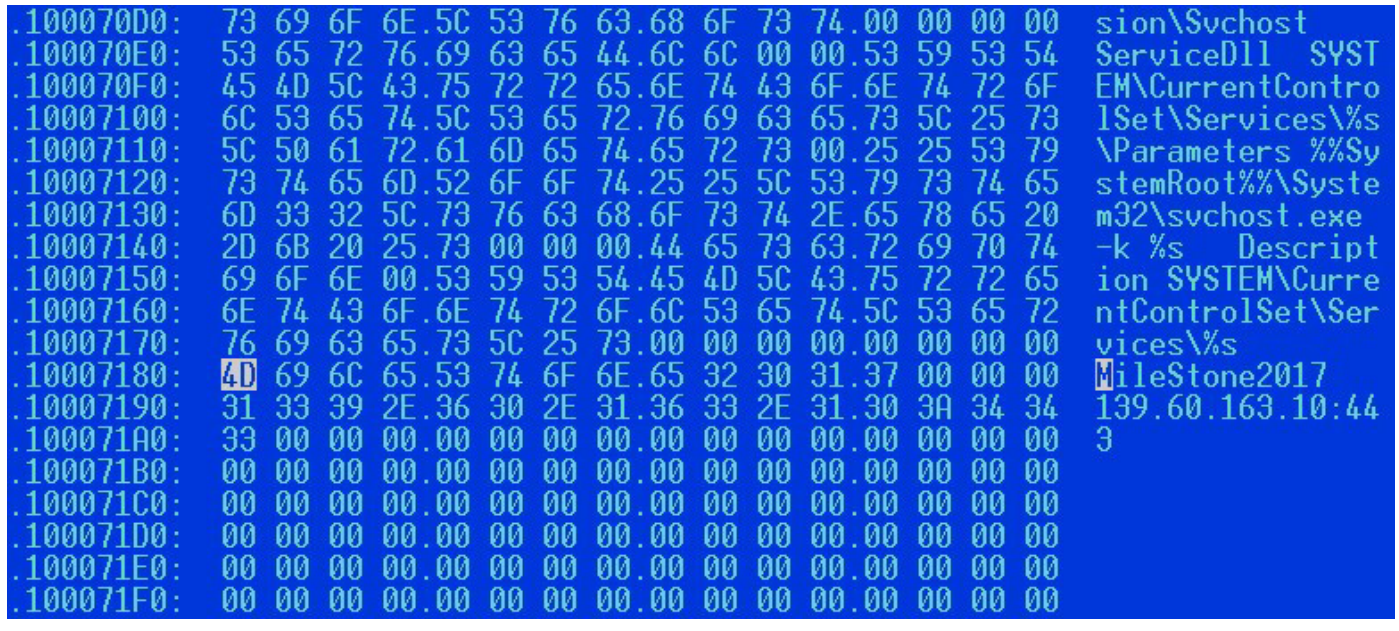
2020년 8월에는 코발트 스트라이크(Cobalt Strike) 의심 샘플이 신고되었다. 파일 이름은 duser.dll였고 코발트 스트라이크는 표적 공격에 악용되기도 해 관련 악성코드 분석을 진행했다. duser.dll 파일은 에러 처리도 하지 않은 API를 호출하는 파일로 보였다.

```
5  GetSystemTimeAsFileTime(0i64);
6  GetProcAddress(0i64, 0i64);
7  ReadFile(0i64, 0i64, 0, 0i64, 0i64);
8  IsValidCodePage(1u);
9  GetVersion();
10 GetCurrentProcess();
11 GetModuleFileNameA(0i64, 0i64, 0);
12 CreateIoCompletionPort(0i64, 0i64, 0i64, 0);
13 QueryPerformanceCounter(0i64);
14 GetStartupInfoW(0i64);
15 GetTickCount();
16 CommConfigDialogA(0i64, 0i64, 0i64);
17 FlsGetValue(0);
18 GetStdHandle(1u);
19 CompareFileTime(0i64, 0i64);
20 GetNativeSystemInfo(lpSystemInfo);
21 AreFileApisANSI();
22 LCMapStringW(0, 0, 0i64, 0, 0i64, 0);
23 BeginUpdateResourceA(0i64, 0);
24 InitializeCriticalSectionAndSpinCount(lpCriticalSection, 1u);
25 GetModuleFileNameW(0i64, 0i64, 0);
26 DeleteCriticalSection(lpCriticalSection);
27 FreeLibrary(0i64);
28 ClearCommBreak(0i64);
29 RtlCaptureContext(ContextRecord);
30 HeapSetInformation(0i64, HeapInformationClass, 0i64, 0i64);
31 BuildCommDCBAndTimeoutsA(0i64, 0i64, 0i64);
32 LoadLibraryW(&LibFileName);
33 SetLastError(1u);
34 IsDebuggerPresent();
35 RtlLookupFunctionEntry(0i64, 0i64, 0i64);
36 SetFilePointer(0i64, 0, 0i64, 0);
```

[그림 16] duser.dll 내부

이후, duser.dll 파일 이름으로 수집된 파일을 분석해서 모두 유사한 형태임을 확인했다. 결과적으로 해당 악성코드는 코발트 스트라이크가 아니라 암호화된 암호화폐 채굴 프로그램을 실행하는 파일로 확인되었다. 그리고, 분석 과정 중 다른 보안 업체에서 공개한 킹 마이너(King Miner) 정보와 일치함을 확인했다.

마지막으로, 2020년 11월 모기업에서 신고 접수된 악성코드는 'MileStone2017' 문자열을 포함한, 전형적인 원격 명령 실행 백도어 악성코드였다.



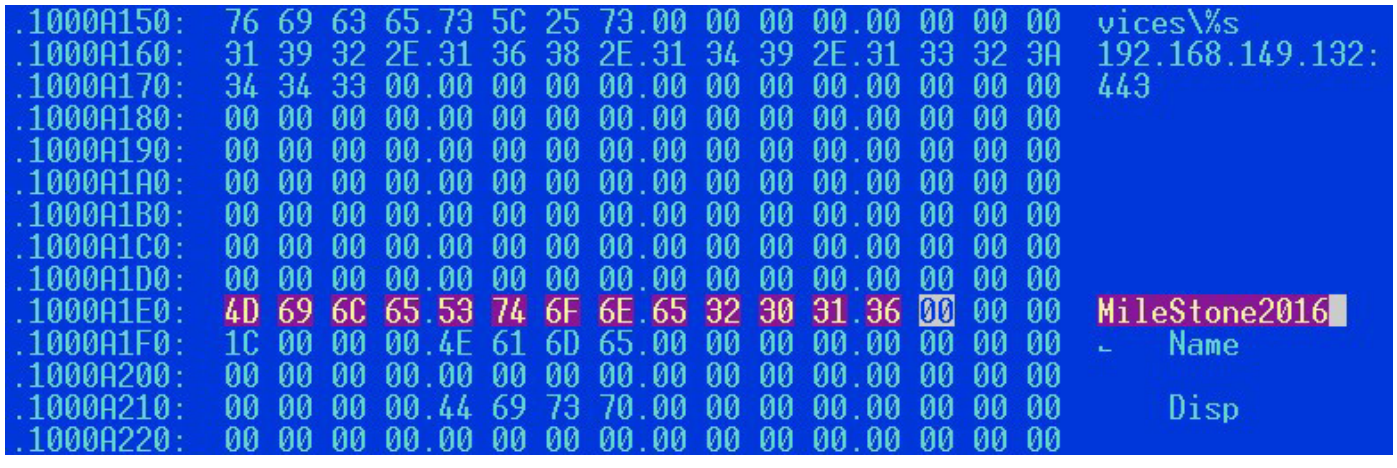
[그림 17] 악성코드의 특징적 문자열

관련 악성코드가 어느 단계에서 발견되었는지 알 수 없지만 해당 기업은 여러 위협 그룹으로부터 지속적인 공격을 받고 있으며, 추가 변형도 몇 가지가 확인되어 추적을 시작했다. 우선 관련 'MileStone2017' 문자열 분석 결과, 이전에 알려진 악성코드는 아닐 가능성이 높았다.

추적을 통해 고객 신고는 11월 초에 접수됐지만, 해당 파일의 최초 수집은 2020년 9월이었으며 모 대학에서도 동일 파일이 수집된 것을 확인했다. 또 해당 파일은 2020년 1월 발견된 변형을 패킹 했다는 점도 알 수 있었다. 공격자가 2017년 제작한 악성코드를 그대로 사용하다 백신 프로그램에 진단된 후, 백신 프로그램 우회를 위해 패커를 사용한 것으로 짐작됐다. 그리고, 공격자가 과거 악성코드를 재활용하고 있음을 확신할 수 있었다.

발견된 몇 가지 변형 익스포트(Export) 함수가 동일하다는 특징을 바탕으로 안랩에서 보유 중인 유사한 형태의 파일 30개를 조사했다. 그리고, 이 중 2016년 수집된 파일에서 'MileStone2016' 문자열이 포함된 초기 버전을 확인했다.

[그림 18] 악성코드의 Export 함수



[그림 19] 악성코드 2016년 버전

악성코드는 특징적 DLL 파일 이름을 사용했고 동일 DLL 파일 이름을 가진 파일을 추가 분석해 'CIA', 'FBI'와 같은 문자열을 포함한 초기 버전이 2011년부터 존재한다는 점도 확인할 수 있었으며 넷봇 어택커(Netbot Attacer)의 변형으로 확인되었다. 넷봇 어택커는 한국에서도 널리 사용된 디도스 공격 도구이며 원격 제어 기능도 포함하고 있다. 관련 소스코드가 공개되어 2010년부터 다양한 변형이 등장했으며 Milestone도 그 변형 중 하나로 확인되었다.

효과적인 위협 대응을 위한 제언

날로 고도화되는 위협 환경에서, 기업의 보안 부서는 적합한 보안 정책을 마련해 보안 사고를 예방하고 침해 발생 시, 신속하고 효과적으로 대응하기 위해 노력하고 있다. 이에, 효과적인 방어 및 대응을 위한 제언 사항을 전하고자 한다.

먼저, 내부 시스템 현황을 파악하고 다양한 보안 솔루션을 상황에 맞게 맞춤화 해야 한다. 행위 기반 솔루션의 경우 내부에서 사용하는 소프트웨어를 설치해 실제 환경과 흡사하게 구성해야 한다. 내부 시스템 가시성 확보, 다양한 로그 장기간 보관, 내부에서 발견된 악성코드 보관 등을

통해 대비하면 침해 사고 발생 시 대응에 도움이 된다. 이상 시스템에서 의심 파일을 찾을 수 있다면, 보안 업체에서 제공하는 정보 수집 도구를 보내고 결과를 받아 커뮤니케이션 시간을 단축할 수 있다.

조직에서 보안 사고가 발생하면 대부분의 보안 부서는 문제가 발생한 시스템의 네트워크를 차단하고 의심스러운 파일을 수집한 뒤 보안 업체에 분석을 의뢰한다. 적극적인 보안 부서는 악성코드 파일을 찾거나, 악성 프로세서를 종료하는 프로그램을 제작하기도 한다. 백신 엔진 업데이트가 나오기 전에는 악성코드의 확산 정도를 확인하기 어렵다. 백신 프로그램의 엔진 업데이트가 제공되면 그 때 악성코드를 치료하거나 시스템을 포맷한다. 하지만, 침해 경로, 확산 정도, 다른 악성코드의 존재 여부를 확인하지 않으면 재감염이 발생할 우려가 있다.

현재의 사이버 공격은 개인 컴퓨터를 비롯해 다양한 경로로 시도된다. 이에, 소위 관문을 지키던 보안 관제, 알려진 악성코드를 진단하는 백신 프로그램을 보완하기 위해 행위 기반 탐지와 가시성 확보를 위한 EDR(Endpoint Detection & Response) 등이 등장했고 일부 제품은 인공지능 탑재를 강조한다.

백신 프로그램은 신·변종 악성코드 대응 역량 측면에서는 부족한 점이 있지만, 알려진 악성코드 위주의 비교적 확실한 악성코드 진단으로 사람의 개입이 적고 효율성이 높다. 반대로, 행위 기반 솔루션은 알려지지 않은 보안 위협을 찾아낼 가능성이 높지만 분석 결과를 최종적으로 판단할 전문가가 필요하다. 행위 기반 솔루션 사용이 증가하면서 필연적으로 악성코드 판단을 의뢰하는 문의도 많아지고 있다.

기업 입장에서는 보안 제품의 가시성을 활용해 내부에서 발생하는 공격 시도와 침해 사고를 즉각적으로 조사하고 관련 악성코드를 분석할 수 있는 분석가가 필요하다. 보안 부서에 분석가가 있으면 가장 좋겠지만, 사정상 분석가가 없다면 보안 업체에 의뢰해야 한다. 제3의 업체는 내부 시스템을 볼 수 있는 권한이 제한적이기 때문에 대규모의 사업을 운영하고, 잦은 공격을 받는 곳이라면 가급적 자체 분석가를 두는 것을 권장한다. 다만, 침해 사고를 대응하고 분석할 수 있는 인력이 확보된 조직에서 조차 사고가 발생하면 대응에 급급해 공격에 대한 분석 정보를 제대로 파악하지 못하는 경우가 빈번하다.

보안 부서의 분석가가 독자적으로 모든 위협 정보를 파악하는 건 물리적으로 굉장히 어려운 일이다. 그렇지만, 본인이 속한 기업과 업종을 주로 공격하는 위협 그룹에 대해서는 공격 방식 등 관련 정보를 상세하게 파악하고 있어야 한다. 따라서, 언론 보도부터 보안 업체 및 기관에서 공개하는 위협 분석 정보 등을 단일 플랫폼에서 제공하는 위협 인텔리전스(TI) 서비스 도입을 고려할 필요가 있다. TI 서비스를 활용하면 다양한 위협 정보를 수집하고 자사에서 발생하는 침해 사고와 악성코드 정보를 효율적으로 비교·분석해 대응 방안을 마련할 수 있다.

사이버 보안 침해를 완벽히 방어하는 것은 불가능하다. 특히, 신뢰 받는 프로그램을 통한 공격을 예방하는 것은 더욱 어렵다. 2020년 12월 확인된 솔라윈즈(SolarWinds) 제품을 통한 공급망 공격 (Supply Chain Attack)처럼, 1년 가까이 들키지 않는 경우도 있다. 따라서, 완벽에 '가까운' 방어 체계를 구축하되, 예방 및 대응 체계를 갖추고 침해 사고가 발생하면 빠르게 복원할 수 있는 능력을 키워야 한다.

결론

공격자는 금전 혹은 정보 탈취라는 명확한 목적을 가지고 지속적으로 공격을 시도한다. 다만, 공격자도 사람인만큼 익숙한 공격 방식과 악성코드를 선호하는 경향이 있다. 따라서, 방어자 입장에서는 공격자들의 특성과 패턴을 분석하고 대응 방안을 마련할 수 있다.

위협 대응의 관점에서 보면, 보안 부서의 역할은 보안 제품 및 장비 운영, 보안 업체와 커뮤니케이션뿐 아니라 자체적인 침해 사고 대응, 악성코드 분석, 관련 공격자에 대한 특징 파악까지 점점 확대되고 있다. 이에 따라, 물리적인 자원의 한계를 극복하고 효율성을 제고하기 위해 TI 서비스 활용을 고려해볼 필요가 있다.

아울러, 기업의 특수한 내부 환경에서 사용되는 플랫폼과 프로토콜 등에 대한 정보를 보안 업체와 유기적으로 공유해야 더 큰 피해를 막을 수 있다. 이제, 기업이 확대되는 보안 부서의 역할을 고려해 역량 강화를 위한 방안을 마련하고, 보안 업체와 효과적으로 협력할 수 있도록 지원해야 한다.