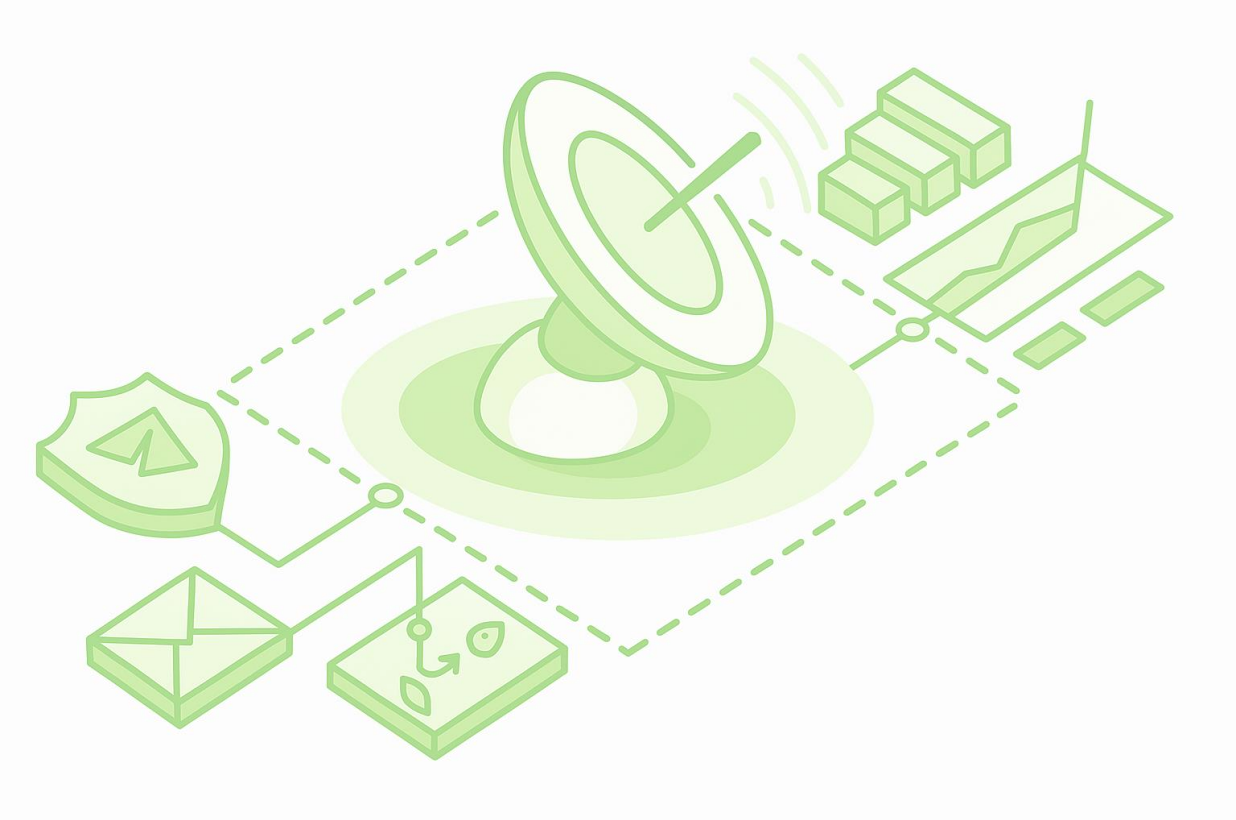
SGA 보안위협 리포트

보안레이더

2026.03



sga 에스지에이솔루션즈(주)

SGA보안레이더 발행본 다운로드

썸네일을 클릭하시면 앞서 발행된 SGA보안레이더를 다운받으실 수 있습니다.

2025.06

주요이슈

BPF Door

sga 에스지에이이피에스(주)

2025.07

주요이슈

SwaetRAT

sga 에스지에이이피에스(주)

2025.08

주요이슈

링크파일 활용한 백도어 악성코드

sga 에스지에이이피에스(주)

2025.09

주요이슈

Gunra Ransomware

sga 에스지에이이피에스(주)

2025.10

주요이슈

LockBit 4.0 랜섬웨어

sga 에스지에이이피에스(주)

2025.11

주요이슈

RokRAT

sga 에스지에이이피에스(주)

2025.12

주요이슈

HybridPetya

sga 에스지에이이피에스(주)

2026.01

주요이슈

DarkCloud Stealer

sga 에스지에이솔루션즈(주)

2026.02

주요이슈

문서위장 악성파일

sga 에스지에이솔루션즈(주)

CONTENTS

발행일자: 2026년 3월

1. 2026년 2월 보안 동향

2026년 2월에도 해킹 사건이 다수 발생한 것으로 파악되었다.

러시아 정부의 지원을 받는 해킹 그룹 APT28, 서유럽 및 중부 유럽 기관을 겨냥한 사이버 공격 진행

러시아 정부의 지원을 받는 것으로 알려진 해킹 그룹 APT28이 서유럽 및 중부 유럽 기관을 겨냥한 사이버 공격 매크로메이즈 작전을 진행한 것으로 확인된다.

공격은 스피어 피싱 이메일로 시작하며, 첨부된 문서에 보이지 않은 추적용 이미지가 들어있고, 사용자가 문서를 열면 이미지를 불러오기 위한 아웃바운드 HTTP 요청이 발생한다. 이를 통해 공격자는 수신자의 문서 열람 여부를 확인하는 비컨 메커니즘이 가동되며 이후 시스템 거점 확보 및 추가 페이로드 전송을 위한 악성 매크로가 실행되는 방식으로 진행된다.

초기 버전은 백그라운드에서 실행되는 헤드리스 브라우저 방식을 사용하였으며, 최신 버전에서는 보안 경고 창을 우회하기 위해 키보드 시뮬레이션 기법을 도입하는 등 보안 탐지를 피하기 위한 기법이 사용되었다. 실행된 매크로는 VBScript와 배치 파일을 사용해 작업 스케줄러를 등록하고 지속성을 유지하며, 엣지 브라우저를 백 그라운드로 실행해 Base64 인코딩 된 HTML 페이로드를 디코딩하고 실행 결과를 수집한다.

북한의 해커 조직인 라자루스, 서비스형 랜섬웨어 메두사 이용한 미국 의료 기관과 중동 기업 공격

북한의 해커 조직인 라자루스가 서비스형 랜섬웨어 메두사를 활용해 미국 의료 기관과 중동 기업을 공격한 것으로 확인된다.

공격 인프라에서는 기성 랜섬웨어와 함께 라자루스 전용 도구들이 다수 발견되어 공격 배후가 특정되었으며, 권한 탈취를 위한 미미카츠를 비롯한 라자루스가 개발한 커백커 백도어, 블라인딩캔 RAT가 사용되었다. 또한 정보 탈취 도구인 인포훅과 브라우저에 저장된 비밀번호를 탈취하는 크롬 스틸러를 사용한 것으로 확인된다.

이는 과거 안다리엘과 같은 산하 조직이 마우이나 홀리고스트 등의 자체 개발한 랜섬웨어를 주로 사용한 것과 달리 기성 RaaS 플랫폼을 채택하는 실용주의로 바뀐 것으로 볼 수 있다.

2. 악성코드 통계 및 분석

2026년 2월 한 달 동안 탐지된 악성코드를 확인한 결과 **총 104,646건의 악성코드가 확인**되었다. 가장 많이 탐지된 악성코드 유형은 Trojan으로 확인되었고, 그 뒤를 Virus, Exploit 형태의 악성코드 유형이 차지하였다. **1월과 비교해 보면 Application.Hacktool, Induc.A 탐지 비율이 증가**하였다.

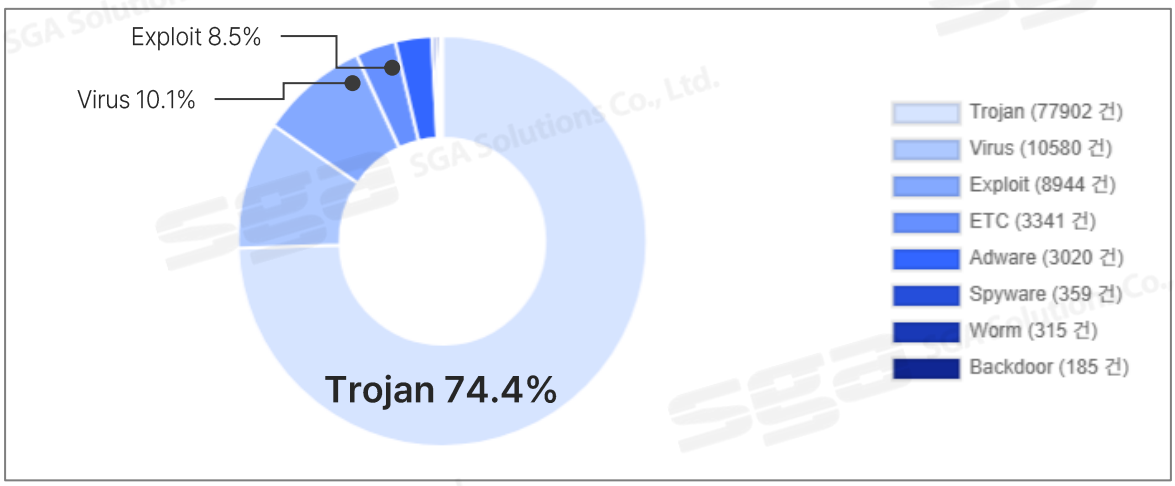
자사에 수집된 피싱 메일은 181건이었다. 사용자를 속이기 위해 가장 많이 사용된 메일 본문 유형은 '인증 및 보안 조치 요구' 였으며, 가장 많이 수집된 피싱 메일 공격 유형은 악성 URL이 첨부된 하이퍼링크 형태의 피싱 메일이었다.

■ 유형별 탐지 통계

2026년 2월 한 달 동안 탐지된 악성코드의 유형을 확인한 결과 **Trojan 형태의 악성코드가 186,622건(74.44%)으로 1순위를 차지**했다. Trojan 형태의 악성코드 중 Ransomware의 탐지 비율은 여전히 높았고, 지난 1월에는 탐지 되지 않았던 Trojan.Cryxos가 탐지 되었다.

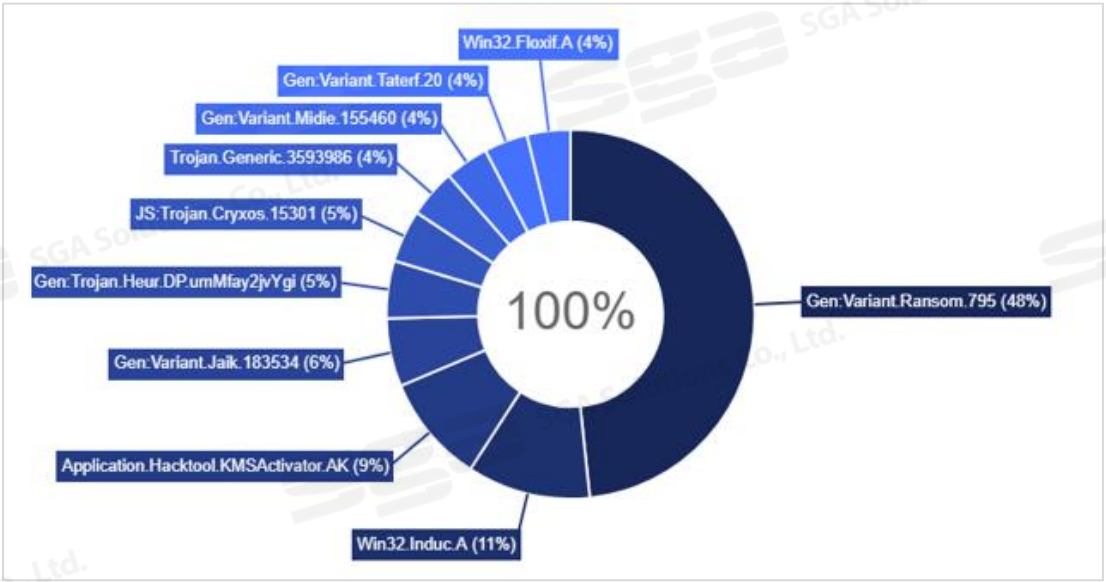
그 다음으로 자기 스스로는 행동할 수 없고, 정상 프로그램에 기생하여 실행되는 **Virus 형태의 악성코드가 10,580건(10.11%)으로 2위를 차지**했다. Virus 형태의 악성코드 중 Induc.A의 탐지 비율이 높았다.

마지막으로 컴퓨터의 소프트웨어나 하드웨어 관련 제품의 버그, 보안 취약점 등 설계상 결함을 이용해 공격하는 **Exploit 형태의 악성코드가 8,944건(8.54%)으로 3위를 차지**했다.Exploit 형태의 악성코드인 Hacktool이 지속적으로 높게 탐지되었고, KMSActivator가 꾸준히 탐지가 되고 있다.



[2026년 2월 유형별 탐지 통계]

■ 악성코드 TOP 10 탐지 통계



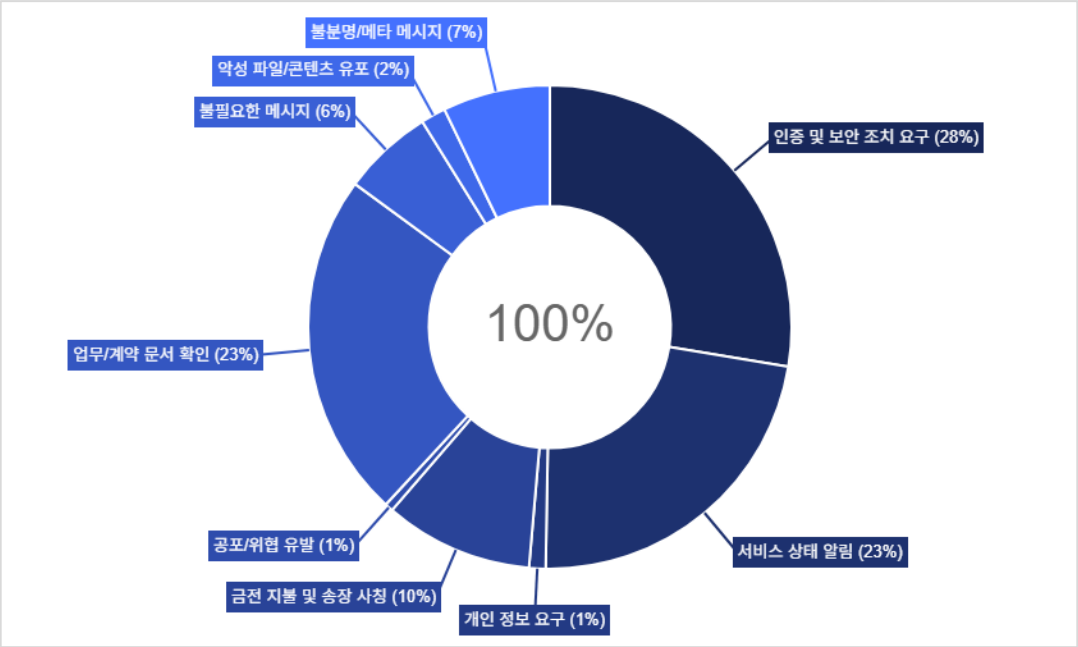
[2026년 2월 악성코드 TOP 10 탐지 통계]

2026년 2월 탐지된 악성코드를 TOP 10으로 통계를 내어 확인한 결과 사용자 PC의 파일을 암호화하여 사용자가 사용할 수 없게 만들며 암호화를 풀어주는 조건으로 금전을 요구하는 악성 소프트웨어의 진단명인 **Ransom.795가 1위를 차지**했다.

델파이의 특정 라이브러리가 감염된 후 컴파일 과정에서 생성되는 EXE 및 DLL 등에 바이러스 코드가 삽입되어 악성 행위를 하는 악성코드 진단명인 **Win32.Induc.A가 2위로 차지**했다.

소프트웨어 불법 인증 도구에 사용되는 악성코드 진단명인 **Application.Hacktool이 3위**를 차지했다. Application.Hacktool은 유료 프로그램을 불법으로 사용할 수 있고, 잠재적으로 프로그램에 악성코드가 심어질 확률이 높아 백신에서 탐지하고 있다.

■ 피싱 메일 본문 및 공격 유형 통계



[2026년 2월 악성코드 피싱 메일 본문 유형 통계]

2026년 2월 한 달 동안 자사에 수집된 피싱 메일은 181건이며, 그 가운데 **"인증 및 보안 조치 요구" 내용이 담긴 피싱 메일이 50건(27.62%)으로 가장 많이 수집**되었다. 비밀번호 만료, 보안 정보 업데이트가 필요하여 인증 및 보안 조치에 대한 내용을 담은 메일이 많았으며, 그 외에 문서를 열람하기 위한 인증 요구 내용이 많았다.

그 다음으로 **"업무/계약 문서 확인" 내용을 담은 피싱 메일이 42건(23.20%)으로 수집**되었다. 메일 내용은 대다수 세금 계산서나 소프트웨어 구입을 위한 견적서 내용이 담긴 메일이 많았다.

또한, 첨부 파일이 포함된 피싱 메일은 53건이 수집되었으며, 첨부 파일의 종류는 PE 파일 5건, Script 파일 32건, 그 외 16건으로 수집되었다.

수집된 악성코드 중 대부분이 Script와 문서 파일을 이용해 사용자를 속이는 형태의 악성코드였으며, 최종적으로는 특정 영역을 클릭하도록 유도하는 형태였다. 이러한 악성코드 안에는 로그인을 유도하기 위한 **악성 도메인 또는 추가 악성 행위를 위해 다운로드하는 URL이 담겨 있었다.**

이처럼 사용자로 하여금 클릭을 할 수 밖에 없는 내용을 담은 주제로 작성되어 있어 사용자의 주의가 필요하다.

3. 악성코드 분석

자사에 수집된 샘플 중 2025년 12월에 폴란드 에너지 기업들을 대상으로 공격에 사용되는 다이노와이퍼(DynoWiper) 악성코드가 수집되었다.

수집된 샘플에 대해서 다음과 같이 분석을 진행하였다

■ 개요

와이퍼(Wiper) 악성코드는 금전적인 이득을 목적으로 하는 랜섬웨어와 다르게 대상 시스템의 데이터를 손상시키거나 삭제하여 시스템 운영을 마비시키는 것을 목적으로 한다.

2021년대 이후 중동 및 동유럽 지역을 중심으로 국가 배후의 해킹 그룹에 의해 주기적으로 수행되고 있는 것이 관측되었으며, 정부 시스템 마비 등을 목적으로 활용되고 있다. 대표적인 와이퍼 형태는 파일 삭제형, 디스크 와이퍼, MBR 삭제, 데이터 베이스 파괴형으로 구분되며, 혼합형이 발견되기도 한다.

분석에 사용된 샘플은 해외 보안 업체인 ESET 연구소에서 발견한 것으로 샌드웜(SandWorm) APT 그룹이 2025년 12월에 폴란드 에너지 기업을 대상으로 공격에 사용된 다이노와이퍼(DynoWiper) 악성코드이다.

다이노와이퍼(DynoWiper)는 샌드웜 APT 그룹이 사용했던 데이터 삭제형 악성코드와 그룹 정책 배포 스크립트에서 유사점이 발견되었기 때문에 샌드웜(SandWorm) APT 그룹이 사용한 것으로 발표하였다.

분류	파괴 대상 및 특징
파일(데이터) 파괴형	특정 확장자나 경로의 파일을 무단 삭제 또는 덮어쓰기 수행
디스크 파괴형	물리적 디스크의 섹터 자체를 파괴하여 데이터 복구 원천 차단
MBR 파괴형	마스터 부트 레코드(MBR)를 오염시켜 시스템 부팅 자체를 불가능하게 함
데이터 베이스 파괴형	기업 및 기관의 핵심 자산인 DB 데이터를 타겟으로 삭제 및 변조 수행

■ 주요 기능

- 파일 손상
- 시스템 파괴
- 강제 재부팅

■ 상세 분석

다이노와이퍼(DynoWiper)

- 다이노와이퍼 기능 확인

```
void Main(void)

{
  HANDLE ProcessHandle;
  BOOL BTokenHandle;
  DWORD DesiredAccess;
  HANDLE *TokenHandle;
  undefined1 auStack_13c8 [4];
  HANDLE hTokenHandle;
  _TOKEN_PRIVILEGES newState;
  uint local_13b0 [1257];
  uint local_c;

  local_c = DAT_004275c0 ^ (uint)auStack_13c8;
                    /* 파일 손상시 사용할 난수 16바이트 생성 */
  Mt19937_seed_default(local_13b0);
  Mt19937_Seed_FillWipePattern16(local_13b0);
                    /* 파일 손상 */
  WipeAllTargetDrives();
                    /* 파일 삭제 */
  DeleteFilesOnAllTargetDrives();
  TokenHandle = &hTokenHandle;
  DesiredAccess = 0x28;
  ProcessHandle = GetCurrentProcess();
  BTokenHandle = OpenProcessToken(ProcessHandle,DesiredAccess,TokenHandle);
  if (BTokenHandle != 0) {
                    /* 권한 상승 */
    newState.PrivilegeCount = 1;
    LookupPrivilegeValueW((LPCWSTR)0x0,L"SeShutdownPrivilege",&newState.Privileges[0].Luid);
    newState.Privileges[0].Attributes = 2;
    AdjustTokenPrivileges(hTokenHandle,0,&newState,0,(PTOKEN_PRIVILEGES)0x0,(PDWORD)0x0);
    CloseHandle(hTokenHandle);
  }
                    /* PC 재부팅 */
  ExitWindowsEx(6,0x20003);
  FUN_00407660(local_c ^ (uint)auStack_13c8);
  return;
}
```

[주요 함수 로직]

분석에 사용된 샘플인 다이노와이퍼의 핵심 기능은 네 가지로 확인되며, 파일 손상에 필요한 난수를 메르센 트위터 혹은 Mt19937으로 알려진 알고리즘을 사용하여 16바이트 값에 대해 초기화 및 생성한다.

파일을 손상시키기 위해 피해 PC에서 검색된 드라이브로부터 제외 폴더를 뺀 전체 파일을 손상을 시키며, Mt19937 난수로 생성된 16바이트는 파일 데이터 손상시키는데 사용된다.

그 후 검색된 드라이브 경로에 있는 파일을 삭제하는 기능을 실행하며 마지막으로 파일 데이터 손상 및 삭제가 완료되고 강제 재부팅이 실행된다.

다이노와이퍼의 진행 순서는 다음과 같다.

순서	내용
1	파일 손상을 위한 16바이트 Mt19937 난수 초기화 및 생성
2	검색되는 드라이브로부터 전체 파일 Mt19937 난수를 이용하여 데이터 손상 (제외 목차 존재)
3	검색되는 드라이브 최상위 경로의 파일 삭제
4	권한 상승 및 피해 PC 강제 재부팅

- 파일 손상 대상 검색

```
local_2ec[0] = (ushort *****)pppppuVar6;
          /* 파일인지 디렉터리인지 FILE_ATTRIBUTE_DIRECTORY(0x10) 값 비교 */
if (((byte)lpFindFileData.dwFileAttributes & 0x10) == 0) {
          /* 파일이면 데이터 손상 시작 */
  PartialEncryptFile((LPCSTR)local_48);
}
else {
          /* 제외 디렉터리 목차 */
  uVar2 = std_wstring_compare(local_30,pppppuVar5,local_20,(ushort *)L"system32",8);
  if ((((((uVar2 != 0) &&
         (uVar2 = std_wstring_compare(local_30,extraout_ECX_00,local_20,
                                      (ushort *)L"windows",7), uVar2 != 0)) &&
        (uVar2 = std_wstring_compare(local_30,extraout_ECX_01,local_20,
                                     (ushort *)L"program files",0xd), uVar2 != 0)) &&
       ((bVar7 = std_wstring_equals(local_30,(ushort *)L"program files(x86)"), !bVar7 &&
         (bVar7 = std_wstring_equals(local_30,(ushort *)L"temp"), !bVar7)))) &&
      ((bVar7 = std_wstring_equals(local_30,(ushort *)L"recycle.bin"), !bVar7 &&
        ((bVar7 = std_wstring_equals(local_30,(ushort *)L"$recycle.bin"), !bVar7 &&
          (bVar7 = std_wstring_equals(local_30,(ushort *)L"boot"), !bVar7)))))) &&
     ((bVar7 = std_wstring_equals(local_30,(ushort *)L"perflogs"), !bVar7 &&
       ((bVar7 = std_wstring_equals(local_30,(ushort *)L"appdata"), !bVar7 &&
         (bVar7 = std_wstring_equals(local_30,(ushort *)L"documents and settings"), !bVar7))
       )))) {
          /* 재귀 진입: 하위 디렉터리 내부를 동일한 방식으로 검색
             및 파일 손상 */
    RecursiveDirectoryWipe((LPCSTR)local_48);
  }
```

[정상 PDF 내용 확인]

파일을 손상시키기 위한 대상 검색 순서는 드라이브 확인, 폴더와 파일 검색, 제외 폴더 확인 순으로 진행한다.

드라이브 검색하기 위해 GetLogicalDrives API을 사용하여 드라이브를 열거하고, GetDriveTypeW를 사용하여 대상을 확인한다. 이때 확인하는 값은 드라이브 속성 타입을 확인하며, DRIVE_REMOVABLE(이동식 드라이브)와 DRIVE_FIXED(고정 드라이브)을 대상으로 확인한다.

그 다음, 폴더와 파일을 검색할 수 있는 API인 FindFirstFileW와 FindNextFileW를 사용하며, 재귀 함수를 통해 내부 폴더 및 파일을 추가적으로 검색한다. 파일과 폴더는 FindFirstFileW와 FindNextFileW에 사용되는 lpFindFileData 값이 0x10(FILE_ATTRIBUTE_DIRECTORY)이면 폴더로 확인하여 구별한다.

다이노와이퍼의 제외 대상 폴더는 11개이며, 시스템 관련 폴더로 확인된다.

제외 대상 목차는 파일 손상 기능에 대한 안정성을 위해 제외한 것으로 확인되며, 제외 목차는 다음과 같다.

내용			
system32	windows	program files	program files(x86)
temp	recycle.bin	$recycle.bin	boot
perflogs	appdata	documents and settings	

- 파일 덮어 씌우기

[파일 손상을 위한 API 흐름]

파일 손상을 진행하기 위해 속성 값을 변경할 수 있는 SetFileAttributesW API를 사용하여 파일의 속성 값을 0x80(FILE_ATTRIBUTE_NORMAL)으로 변경한다. 파일의 속성 값 0x80은 파일이 숨겨져 있거나, 시스템 파일, 읽기 전용 등의 속성일 때, 일반 파일 속성으로 초기화하기 위해 사용된다.

일반 파일로 초기화된 파일은 생성한 16바이트의 Mt19937 난수가 파일 헤더 데이터에 덮어쓰며 이 때, 파일 입출력 API를 사용하여 파일의 크기와 상관없이 덮어쓴다.

그 후 GetFileSize로 파일 크기를 16바이트보다 큰지 확인하고 파일의 크기가 크다면 BuildRandomOffsets 함수 내에서 파일 크기를 16으로 나눈다.

여기서 나온 몫의 1% 값으로 데이터를 덮어 씌울 횟수를 설정하며 최대 4096로 제한한다. 덮어쓸 파일의 오프셋은 BuildRandomOffsets 함수에서 랜덤 한 값으로 설정된다.

이 설정 값을 이용하여 Mt19937 난수 값과 파일 입출력 API인 SetFilePointerEx와 WriteFile으로 횟수만큼 데이터를 손상시킨다.

- 드라이브 최상위 경로 파일 삭제

```
                /* ".", ".." 제외 */
    if (nameDiffFlag != 0) {
      puVar3 = WString_ConcatWStringAndCStrW(local_2e4,local_78,(int *)&\);
      local_8._0_1_ = 2;
                /* 경로 조합 */
      BuildPathW((int *)entryFullPathW,puVar3,(int *)lpFindFileData.cFileName);
      local_8._0_1_ = 4;
      if (7 < local_2d0) {
        FID_conflict:_free(local_2e4[0]);
      }
      local_2d0 = 7;
      local_2e4[0] = (void *)((uint)local_2e4[0] & 0xffff0000);
      local_2d4 = 0;
      ConvertWStringToUtf8((undefined1 *)entryFullPathUtf8,(LPCWSTR)entryFullPathW);
      local_8 = CONCAT31(local_8._1_3_,5);
                /* FILE_ATTRIBUTE_DIRECTORY (0x10), 폴더에 해당 */
      if (((byte)lpFindFileData.dwFileAttributes & 0x10) != 0) {
        pSearchPatternW = entryFullPathW;
        if (7 < local_1c) {
          pSearchPatternW = (LPCWSTR ****)entryFullPathW[0];
        }
                /* FILE_ATTRIBUTE_NORMAL (0x80): 모든 속성(읽기 전용, 숨김, 시스템
                   등)을 해제하고 "아무 속성도 없는 일반 상태"
                 */
        SetFileAttributesW((LPCWSTR)pSearchPatternW,0x80);
        RecursiveDirectoryWipe((LPCSTR)entryFullPathUtf8);
      }
      pSearchPatternW = entryFullPathW;
      if (7 < local_1c) {
        pSearchPatternW = (LPCWSTR ****)entryFullPathW[0];
      }
      DeleteFileW((LPCWSTR)pSearchPatternW);
      if (0xf < local_4c) {
        FID_conflict:_free(entryFullPathUtf8[0]);
      }
      local_8 = CONCAT31(local_8._1_3_,1);
      local_4c = 0xf;
      local_50 = 0;
      entryFullPathUtf8[0] = (void *)((uint)entryFullPathUtf8[0] & 0xffffff00);
      if (7 < local_1c) {
        FID_conflict:_free(entryFullPathW[0]);
      }
    }
  }
  nextFile = FindNextFileW(hFindFile,&lpFindFileData);
} while (nextFile != 0);
FindClose(hFindFile);
```

[파일 삭제 로직]

파일 손상 기능이 완료된 후 검색된 드라이브 최상위 경로의 파일 삭제를 진행한다.

시스템 파괴를 하기 위해 GetLogicalDrives와 GetDriveTypeW API를 사용하여 이동식 드라이브와 고정 드라이브를 검색한다.

파일 손상 기능과 다르게 파일을 삭제하는 조건은 존재하지 않으며 대상이 파일이 아닌 폴더로 확인되면 파일 손상 기능을 한 번 더 수행하게 된다.

그 후 파일 입출력 API인 DeleteFileW를 사용하여 드라이브 최상위 경로 내 파일들을 삭제를 진행한다.

삭제되는 파일들 중 시스템 부팅 과정에서 사용되는 BOOTNXT 파일이 포함되어 있으며, 시스템 재부팅 시 시스템 파괴가 된 것을 확인할 수 있다.

- 시스템 파괴 결과

[자동 복구 확인]

ExitWindowsEx을 사용하여 시스템 종료, 재부팅, 전원 끄기를 진행하려면 호출하는 프로세스에 권한(SeShutdownPrivilege)이 부여되어 있어야 한다.

이 권한은 비활성화되어 있는 경우가 있으며, AdjustTokenPrivileges API를 사용하여 활성화를 진행한다.

권한 활성화가 진행되면, ExitWindowsEx를 사용하여 시스템이 종료되고 다시 시작한다.

재부팅이 진행될 때, 부팅 관련 BOOTNXT 파일이 삭제되어 있어 정상적인 부팅이 아닌 자동 복구 화면이 출력이 된다.

4. 주요 보안 뉴스

北 라자루스, '메두사' 랜섬웨어 도입해 미국·중동 동시 타격

북한 연계 해커 조직인 라자루스(Lazarus)가 서비스형 랜섬웨어(RaaS) '메두사'(Medusa)를 활용해 미국 의료 기관과 중동 기업을 공격한 정황이 확인됐다.

- 출처: https://www.boannews.com/media/view.asp?idx=142388&page=1&kind=1

러시아 배후 APT28, 서유럽 타깃 사이버 공격 '매크로메이즈 작전' 포착

러시아 정부의 지원을 받는 것으로 알려진 해킹 그룹 APT28이 서유럽 및 중부 유럽 기관을 겨냥한 사이버 공격 캠페인 '매크로메이즈 작전'(Operation MacroMaze)을 전개한 것으로 확인됐다.

- 출처: https://www.boannews.com/media/view.asp?idx=142371&page=1&kind=4

SGA 보안위협 리포트
보안레이더

SGA 에스지에이솔루션즈(주)
https://www.sgasol.kr
경기도 의왕시 광진말로 54, 의왕 스마트시티퀀텀 B동 5층 525호