PIOLINK

Sinobi Ransomware

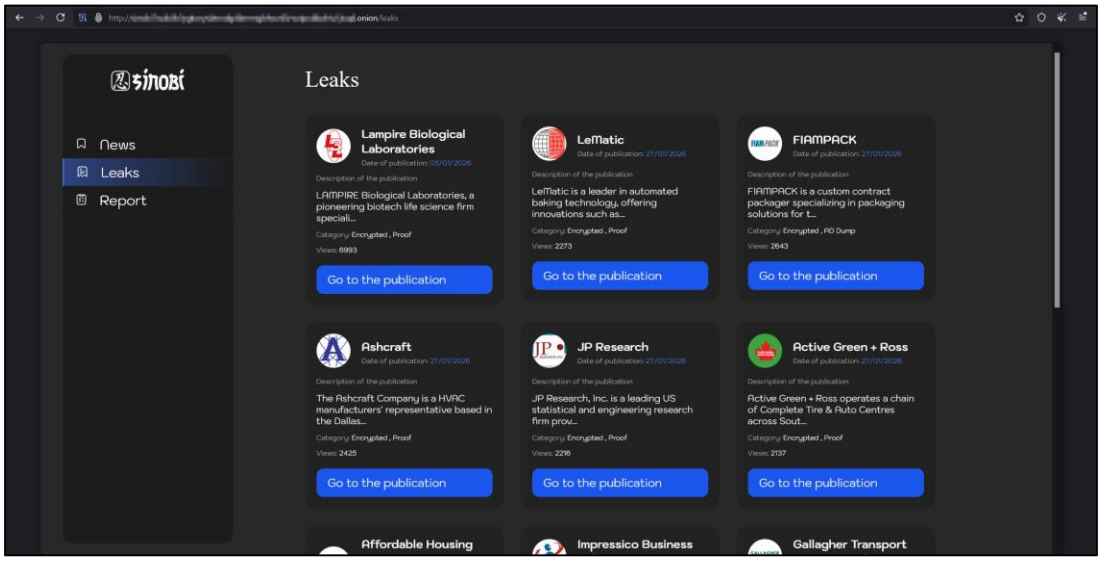
INC, Lynx의 계보를 잇는 폐쇄형 하이브리드 RaaS 그룹

사이버위협분석팀

• Sinobi Ransomware

1) 개요

Sinobi Ransomware는 2025년 6월에 최초로 등장한 이후 2달도 채 되지 않아 전체 랜섬웨어 사건의 10% 이상을 차지하며 급속도로 세력을 확장한 하이브리드 RaaS 그룹이다. Sinobi의 기술적 계보는 2023년 8월에 등장한 INC Ransomware로부터 시작된다. 2024년 5월에 다크웹 포럼에 'salfetka'라는 액터가 INC Ransom의 소스 코드를 매물로 올렸고 이후 2024년 7월에 Lynx Ransomware가 등장했다. 바이너리 분석 결과 함수와 코드 유사도에서 Lynx는 INC Ransom의 기술적 후계자임이 입증되었다. Lynx는 약 1년간 활발히 활동하다가 2025년 중반, 활동이 축소되었고 그와 거의 동시에 Sinobi가 등장했다. 코드 중복, 유출 사이트 인프라의 유사성, 운영 방법론의 공통점 등을 근거로 Sinobi는 Lynx의 리브랜딩으로 평가받고 있다.



<그림 1> Sinobi의 다크웹 유출 사이트

Sinobi는 이중 갈취 전략을 사용한다. 1차로 피해 조직의 파일을 암호화하여 업무를 마비시키고 복호화 대가를 요구하며, 2차로 사전에 탈취한 데이터를 Tor 기반 유출 사이트에 공개하겠다고 협박하여 추가적인 금전적 압박을 가한다.

Sinobi 그룹의 큰 특징은 계열사를 엄격히 통제한다는 것과 선별적 타겟팅을 한다는 것이다. 일반적인 RaaS 그룹이 공개 모집을 통해 계열사를 확대하는 것과 달리, Sinobi는 비공개 네트워크를 통해 추천을 받은 전문가만을 선별적으로 영입한다. 또한 타겟은 미국에 90% 이상이 집중되며 중견 기업을 전략적으로 선정한다. 이는 충분한 랜섬 지불 능력이 있으면서도 대기업 대비 보안 태세가 취약한 조직의 특성을 악용하는 것이다.

<그림 2> Sinobi 공격 흐름도

2) 피해 사례

Central Jersey Medical Center

○ 개요

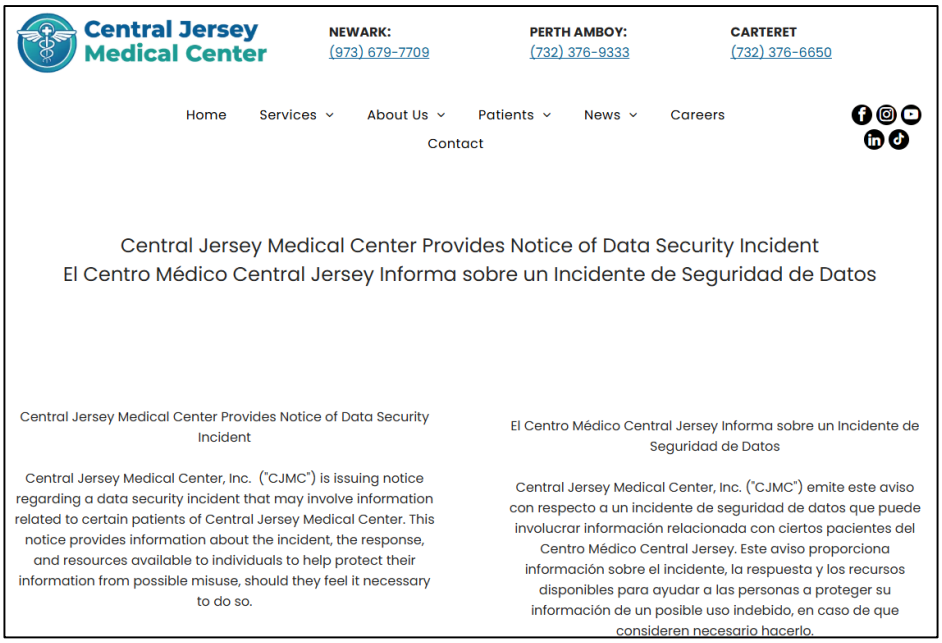
Central Jersey Medical Center(CJMC)는 미국 뉴저지주에 위치한 3개 지역 센터를 운영하는 연방 자격 건강센터로, 지역 사회에 치과 및 일반 의료 서비스를 제공한다. 2025년 8월 25일, 사이버 공격자가 CJMC의 치과 서버 네트워크에 무단 접근하여 랜섬웨어를 배포하고 파일을 암호화했다.

○ 경과

공격이 발생한 후 CJMC는 2025년 10월 13일경 해당 사건을 인지하고 즉시 네트워크 보안 조치 및 사건의 범위 파악을 위한 조사에 착수했다. 조사 결과, 환자 정보가 포함된 파일이 접근되었거나 탈취되었을 가능성이 확인되었다. Sinobi 그룹은 자신들의 Tor 유출 사이트 피해자 목록에 CJMC를 등재했으며 약 930GB의 환자 데이터를 다운로드했다고 주장했다.

○ 피해 규모

유출 가능성이 확인된 데이터에는 인구통계 정보(이름, 생년월일, 주소, 연락처), 사회보장번호, 치과 기록 번호, 건강보험 정보, 치과 진단 및 치료 이력, 청구 정보가 포함되어 있다. 다만 CJMC의 EMR 시스템은 영향을 받지 않았으며, 금융 정보도 침해되지 않은 것으로 확인되었다. 침해 통지서 발송 시점까지 유출된 데이터의 실제 악용 사례는 확인되지 않았다.


Central Jersey Medical Center

NEWARK: (973) 679-7709
PERTH AMBOY: (732) 376-9333
CARTERET (732) 376-6650

Home Services About Us Patients News Careers Contact

Central Jersey Medical Center Provides Notice of Data Security Incident
El Centro Médico Central Jersey Informa sobre un Incidente de Seguridad de Datos

Central Jersey Medical Center Provides Notice of Data Security Incident

Central Jersey Medical Center, Inc. ("CJMC") is issuing notice regarding a data security incident that may involve information related to certain patients of Central Jersey Medical Center. This notice provides information about the incident, the response, and resources available to individuals to help protect their information from possible misuse, should they feel it necessary to do so.

El Centro Médico Central Jersey Informa sobre un Incidente de Seguridad de Datos

Central Jersey Medical Center, Inc. ("CJMC") emite este aviso con respecto a un incidente de seguridad de datos que puede involucrar información relacionada con ciertos pacientes del Centro Médico Central Jersey. Este aviso proporciona información sobre el incidente, la respuesta y los recursos disponibles para ayudar a las personas a proteger su información de un posible uso indebido, en caso de que consideren necesario hacerlo.

<그림 3> Central Jersey Medical Center의 공지 (출처 : cjmc.us)

Watsonville Community Hospital

○ 개요

Watsonville Community Hospital은 미국 캘리포니아주 Watsonville에 위치한 1895년 설립된 106병상 규모의 급성기 의료 시설이다. 이 병원은 2024년 11월 Termite 랜섬웨어 그룹에 의해 최초 침해를 당한 뒤, 2025년 8월 9일 Sinobi 그룹에 의해 추가 공격을 받으며 이중 피해를 입은 이례적 사례를 남겼다.

○ 경과

2024년 11월 최초 침해 시 병원은 사이버 공격으로 인한 시스템 장애를 공시했으나 랜섬웨어 공격임을 확인하지는 않았다. 2024년 12월 11일, Termite 그룹이 자신들의 유출 사이트에 병원을 등재하고 인사 데이터와 환자 데이터를 포함한 증거를 게시했다. 이후 2025년 7월, Termite는 탈취했다고 주장하는 데이터를 유출했다. 그러나 2025년 10월 1일, Sinobi 그룹이 별도로 해당 병원을 자신들의 유출 사이트에 등재했으며, 공격 추정일은 2025년 8월 9일로 기록되었다. Sinobi가 유출한 데이터에는 2025년 3월 이후의 파일들이 다수 포함되어 있어 2024년 11월 Termite 사건과는 별개의 침해일 가능성이 높은 것으로 추측된다.

○ 피해 규모

병원은 2025년 10월 15일자로 캘리포니아주 법무장관실에 환자 통지서를 제출했으나, 정확한 피해 규모는 공개되지 않았다. 이 사례는 한 의료기관이 복수의 랜섬웨어 그룹에 의해 반복적으로 표적이 되는 현실을 보여주며, 초기 침해 후 적절한 보안 강화 조치가 이루어지지 않으면 추가 공격에 노출될 수 있음을 시사한다.

<그림 4> Watsonville Community Hospital의 Sinobi 유출 사이트 등재

Cardiovascular Medical Group of Southern California

○ 개요

Cardiovascular Medical Group of Southern California(CVMG)는 미국 캘리포니아주 남부에 위치한 심장혈관 전문 의료기관으로, 심장병 및 내과 전문의가 진단 검사, 질환 관리, 첨단 치료를 포함한 포괄적인 심장혈관 서비스를 제공한다. 2026년 1월 11일, Sinobi 그룹이 CVMG를 자신들의 다크웹 유출 사이트에 등재하며 공격 사실을 주장했다.

○ 경과

Sinobi 유출 사이트의 게시물에는 CVMG에 대한 침해 주장과 함께 관련 URL이 포함되어 있으나, 유출된 데이터의 구체적 범위, 탈취된 데이터 용량, 랜섬 요구 금액 등 상세 정보는 공개되지 않았다. CVMG는 해당 시점까지 공격에 대한 공식 확인 또는 부인을 하지 않은 상태이다. SOSRansomware는 이 사례를 Sinobi의 의료 부문 공격 강화 전략의 대표적 사례로 분석하며, 심장혈관 전문 의료기관이라는 특성상 환자 데이터의 민감도가 매우 높다는 점을 지적했다.

○ 피해 규모

구체적인 데이터 유출 규모는 미확인 상태이나, 심장혈관 전문 의료기관의 특성상 유출 데이터에는 심장질환 진단 결과, 치료 이력, 약물 처방 정보, 보험 청구 데이터 등 고도의 민감한 의료 정보가 포함되었을 가능성이 높다. 이 사건은 Sinobi가 2026년 초부터 의료 기관에 대한 타겟팅을 더욱 강화하고 있음을 보여주는 사례이며, 전문 클리닉 및 소규모 의료 시설까지 공격 범위를 확대하고 있음을 시사한다.

<그림 5> Cardiovascular Medical Group of Southern California의 Sinobi 유출 사이트 등재

3) Sinobi Ransomware 공격기법 분석

초기 침투

Sinobi Ransomware의 초기 침투는 주로 세 가지 경로를 통해 이루어진다. 첫째, SonicWall 기기의 알려진 취약점인 CVE-2024-40766과 CVE-2024-53704를 이용한다. CVE-2024-40766은 SonicOS 관리 및 SSL VPN 경로의 부적절한 접근 제어 취약점으로, 무단 접근 및 기기 충돌을 유발할 수 있다. CVE-2024-53704는 SSL VPN의 인증 우회 취약점으로, 세션 쿠키의 부적절한 조작을 통해 유효한 자격 증명 없이 원격 접근이 가능하다. 둘째, 관리형 서비스 제공업체의 침해된 자격 증명을 활용한다. 셋째, 초기 접근 브로커를 통해 다크웹 마켓플레이스에서 유효한 VPN 및 원격 접근 자격 증명을 구매하거나 피싱 공격을 통해 자격 증명을 확보한다.

명령줄 인자 파싱

Sinobi Ransomware는 실행 시 공격자가 지정한 다양한 명령줄 옵션을 파싱하여 동작을 결정한다. 이는 RaaS 모델에서 계열사가 피해 환경에 맞게 공격을 세부 조정할 수 있도록 설계된 구조이다.

플래그	기능
--verbose	상세 로그 출력 활성화
--kill	프로세스 및 서비스 강제 종료 (sql, veeam, backup 등 킬 리스트 대상)
--stop-processes	RestartManager를 통한 파일 잠금 프로세스 종료
--encrypt-network	네트워크 공유 드라이브 암호화
--load-drives	숨겨진 드라이브 마운트 후 암호화 (부트로더 손상 가능)
--silent	무흔적 암호화 (.SINOBI 확장자 추가 및 랜섬노트 생성 비활성화)
--safe-mode	안전모드 진입
--hide-cmd	콘솔 창 숨김
--no-background	바탕화면 변경 비활성화
--no-print	프린터 출력 비활성화
--mode	암호화 모드 선택 (fast/medium/slow/entire, 기본값: medium)
--file	특정 파일 암호화 (쉼표로 복수 지정 가능)
--dir	특정 디렉토리 암호화 (쉼표로 복수 지정 가능)
--help	도움말 출력 후 종료

<표 1> 옵션플래그

프로세스 및 서비스 종료

명령줄 인자 파싱이 완료되면, Sinobi는 암호화 효과를 극대화하기 위해 전처리 작업을 수행한다. 우선 프로세스와 서비스를 종료하는데, --kill 플래그가 활성화된 경우, CreateToolhelp32Snapshot API를 사용하여 현재 실행 중인 모든 프로세스를 스냅샷으로 캡처한 뒤, 프로세스 이름에 특정 키워드가 포함되어 있는지 확인하여 해당 프로세스를 강제 종료한다. 서비스 종료는 OpenSCManagerW를 통해 서비스 제어 관리자에 접근한 후, 대상 서비스의 종속 서비스를 재귀적으로 먼저 종료하고, 이후 대상 서비스를 종료하는 방식으로 진행된다. 예를 들어 SQL Server를 종료하려면, 먼저 SQL Agent, SSRS 등 종속 서비스를 순서대로 종료한 뒤 최종적으로 SQL Server 자체를 종료한다.

```
v87 = OpenProcess(1u, 0, pe.th32ProcessID);
v88 = v87;
if ( v87 )
{
  if ( TerminateProcess(v87, 9u) && byte_140033C78 )
    print(L"[+] Proccess %s with PID: %d was killed succesffully\n", pe.szExeFile, pe.th32ProcessID);
  CloseHandle(v88);
}
```

<그림 6> 프로세스 종료 로직 중 일부

데이터베이스와 백업 관련 서비스를 종료하는 이유는, 이 서비스들이 실행 중일 때 해당 파일들은 운영체제에 의해 잠겨 있어 암호화할 수 없기 때문이다. 서비스를 먼저 종료함으로써 파일 잠금을 해제하고 암호화 대상 범위를 확대한다. 특히 Veeam 백업 서비스를 종료하는 것은 피해자의 백업 복구 경로를 사전에 차단하려는 의도이다.

```
.rdata:000000014002DC70                 text "UTF-16LE", 'sql',0
.rdata:000000014002DC78 aVeeam:                                 ; DATA XREF: .rdata:000000014002B9B0↑o
.rdata:000000014002DC78                                         ; .rdata:000000014002B9D8↑o
.rdata:000000014002DC78                 text "UTF-16LE", 'veeam',0
.rdata:000000014002DC84                 align 8
.rdata:000000014002DC88 aBackup:                                ; DATA XREF: .rdata:000000014002B9B8↑o
.rdata:000000014002DC88                                         ; .rdata:000000014002B9E0↑o
.rdata:000000014002DC88                 text "UTF-16LE", 'backup',0
.rdata:000000014002DC96                 align 8
.rdata:000000014002DC98 aExchange:                              ; DATA XREF: .rdata:000000014002B9C0↑o
.rdata:000000014002DC98                                         ; .rdata:000000014002B9E8↑o
.rdata:000000014002DC98                 text "UTF-16LE", 'exchange',0
.rdata:000000014002DCAA                 align 10h
.rdata:000000014002DCB0 aJava:                                  ; DATA XREF: .rdata:000000014002B9F0↑o
.rdata:000000014002DCB0                 text "UTF-16LE", 'java',0
.rdata:000000014002DCBA                 align 20h
.rdata:000000014002DCC0 aNotepad:                               ; DATA XREF: .rdata:000000014002B9F8↑o
.rdata:000000014002DCC0                 text "UTF-16LE", 'notepad',0
```

<그림 7> 종료 대상 프로세스 목록

휴지통 비우기

전처리 단계에서 Sinobi는 SHEmptyRecycleBinA API를 호출하여 휴지통의 모든 파일을 삭제한다. 이는 랜섬웨어 감염 후 피해자가 휴지통에서 파일을 복원하는 것을 방지하기 위함이다. 이 동작은 뒤이어 수행되는 볼륨 섀도 복사본 삭제와 함께 피해자의 파일 복구 경로를 체계적으로 차단하려는 전략의 일부이다. Sinobi가 디렉토리를 암호화할 때 $RECYCLE.BIN 폴더를 암호화 대상에서 제외하는데, 이미 휴지통의 내용물이 삭제되어있기 때문이다.

```
if ( v100 )
  print(L"[+] Recycling bin...\n");
SHEmptyRecycleBinA(0, 0, 7u);
```

<그림 8> Sinobi의 SHEmptyRecycleBinA API 호출

숨겨진 드라이브 마운트 및 VSS 삭제

Sinobi는 일반적으로 사용자에게 보이지 않는 숨겨진 볼륨을 탐색하여 사용하지 않는 드라이브 문자에 마운트한다. FindFirstVolumeW, FindNextVolumeW API로 시스템에 존재하는 모든 볼륨 GUID를 열거하고, 이미 드라이브 문자가 할당된 볼륨은 건너뛴다. 마운트되지 않은 볼륨이 발견되면 사용하지 않는 드라이브 문자에 SetVolumeMountPointW로 마운트하여 암호화 대상에 포함시킨다. 이 기능은 복구 파티션까지 마운트하여 암호화하는데, 이 경우 운영체제의 부트로더가 손상되어 시스템 부팅 자체가 불가능해진다.

또한 Sinobi는 드라이브 암호화 시작 전에 Windows의 볼륨 섀도 복사본(VSS)를 삭제한다. VSS는 Windows가 자동으로 생성하는 파일의 이전 버전 백업으로 랜섬웨어 피해 시 이를 통해 파일을 복원할 수 있기 때문에 공격자 입장에서 반드시 제거해야 하는 대상이다.

```
FirstVolumeW = FindFirstVolumeW(v7, 0x8000u);
do
{
  if ( !v1 )
    break;
  if ( GetVolumePathNamesForVolumeNameW(v7, szVolumePathNames, 0x78u, cchReturnLength)
    && lstrlenW(szVolumePathNames) == 3 )
  {
    szVolumePathNames[0] = 0;
  }
  else
  {
    v9 = lpszVolumeMountPoint[--v1];
    if ( SetVolumeMountPointW(v9, v7) )
    {
      if ( byte_140033C78 )
        print(L"\t[+] Mounted % s\n", v9);
    }
```

<그림 9> 숨겨진 드라이브 마운트 로직 중 일부

암호화 키 생성

Sinobi Ransomware는 Curve25519 타원곡선 디피-헬만 키 교환과 AES-128 대칭 암호화를 결합한 하이브리드 암호화를 사용한다. 이는 현대 랜섬웨어에서 채택하는 가장 진보된 암호화 방식 중 하나이다.

Sinobi는 우선 하드코딩된 공격자의 Curve25519 공개키를 디코딩한다. Base64로 인코딩된 문자열을 CryptStringToBinaryA 함수로 디코딩하면 32바이트의 바이너리 데이터가 된다. 이 공개키는 공격자가 보유한 개인키와 대응되며, 피해자의 파일을 복호화하는데 필수적인 암호를 유도하는 기반이 된다.

```
v5 = lstrlenA("Il2uG3L5S9oCrKxvozNzWSynv8ZWxN09+9w8aLpAhBE=");
if ( !CryptStringToBinaryA("Il2uG3L5S9oCrKxvozNzWSynv8ZWxN09+9w8aLpAhBE=", v5, 1u, 0, &pcbBinary, 0, 0) )
  goto LABEL_11;
v6 = pcbBinary;
ProcessHeap = GetProcessHeap();
v8 = HeapAlloc(ProcessHeap, 0, v6);
if ( !v8 )
  goto LABEL_11;
v9 = lstrlenA("Il2uG3L5S9oCrKxvozNzWSynv8ZWxN09+9w8aLpAhBE=");
if ( !CryptStringToBinaryA("Il2uG3L5S9oCrKxvozNzWSynv8ZWxN09+9w8aLpAhBE=", v9, 1u, (BYTE *)v8, &pcbBinary, 0, 0) )
{
  v10 = GetProcessHeap();
  HeapFree(v10, 0, v8);
```

<그림 10> 공개키 디코딩 로직

또한 Sinobi는 각 파일마다 고유한 임시 키 쌍을 생성한다. CryptAcquireContextW로 암호화 컨텍스트를 생성한 뒤, CryptGenRandom 함수로 32바이트의 난수를 생성한다. 그 후 Curve25519 표준이 요구하는 형식에 맞게 비트를 조정한다. 이를 클램핑이라 하며, 이것이 해당 파일 전용 임시 개인키가 된다. 임시 개인키로 두 번의 타원곡선 연산을 수행한다. 첫 번째 연산으로 임시 공개키를 생성하고 두번째 연산으로 공격자의 공개키와 조합하여 Shared Secret(임시 개인키 * 공격자의 공개키)을 생성한다. Shared Secret을 SHA-512로 해시화하여 64바이트의 값을 얻어서 앞 16바이트를 AES 암호화 키로, 그 다음 16바이트를 초기화 벡터(IV)로 사용한다.

이 구조의 핵심은 공격자만이 자신의 개인키와 파일에 저장된 임시 공개키를 이용해 Shared Secret을 계산할 수 있다는 점이다. 제3자는 어느 쪽의 개인키도 모르기 때문에 복호화가 불가능하다.

```
if ( v6 )
{
  if ( CryptGenRandom(phProv[0], 0x20u, v6) )
  {
    CryptReleaseContext(phProv[0], 0);
  }
  else
  {
    CryptReleaseContext(phProv[0], 0);
    v7 = GetProcessHeap();
    HeapFree(v7, 0, v4);
    *(a1 + 40) = GetLastError();
    v4 = 0;
  }
}
```

<그림 11> 파일별 임시 키 생성 로직

파일 암호화

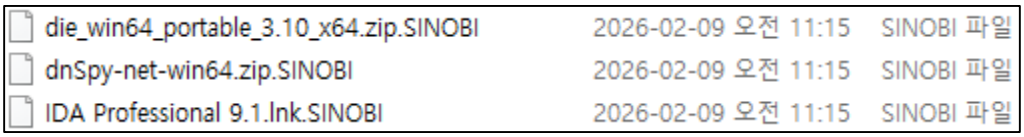
Sinobi Ransomware는 암호화 전에 대상 파일들의 읽기 전용 속성을 먼저 해제한다. 다른 프로그램이 파일을 사용 중일 경우, 두 단계로 잠금 해제를 시도한다. 우선 RestartManager API로 파일을 잠그고 있는 프로세스를 찾아 종료한다. 이것으로 해결되지 않으면 파일의 소유권을 강제로 탈취한다. 모든 사용자에게 전체 접근 권한을 부여하여 파일을 열 수 있도록 만든다.

Sinobi는 Windows의 IOCP를 활용하여 여러 파일을 동시에 암호화한다. CPU 코어 수 * 4개의 워커 스레드가 병렬로 동작하며, 동시 작업 수가 100개를 넘으면 대기하여 시스템 과부하를 방지한다. 각 워커 스레드는 아래 <표 2>와 같이 상태를 순서대로 처리한다.

상태	동작
상태 0	암호화할 파일 구간 계산 (모드에 따라 일부 구간 건너뜀)
상태 1	AES-128-CTR로 데이터 암호화
상태 2	암호화된 데이터를 파일에 기록
상태 3	파일 끝에서 "SINOBI" 마커 확인 (이미 암호화된 파일이면 건너뜀)
상태 4	116바이트 푸터 기록
상태 5	.SINOBI 확장자 추가, 랜섬노트 생성, 핸들 종료 (--silent 시 생략)
상태 6	오류 처리
상태 7	스레드 종료

<표 2> IOCP 워커 스레드 상태 머신

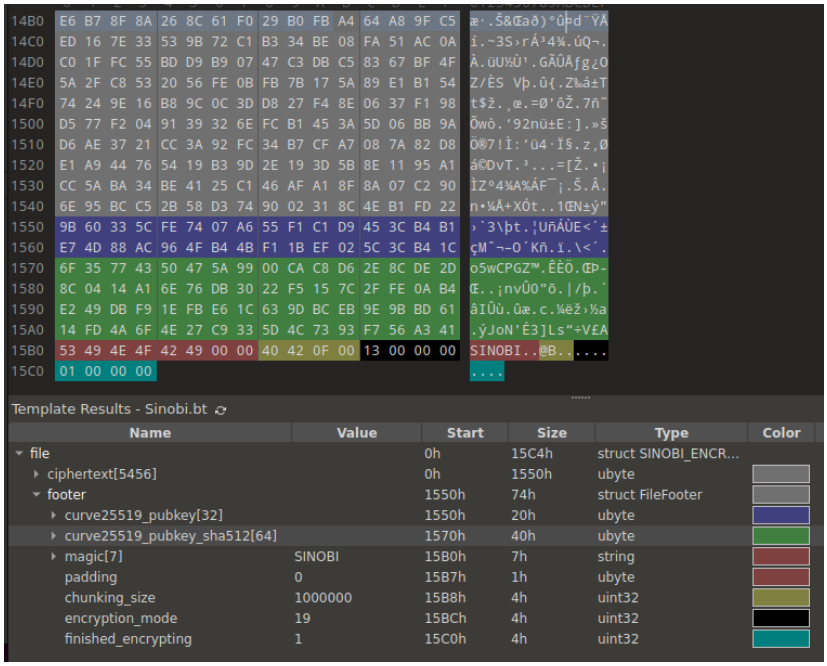
암호화 대상 파일들은 약 1MB 단위의 청크로 나뉘어 암호화된다. fast 모드라면, 1MB만 암호화하고 19MB는 건너뛰어 파일의 약 5%만 암호화하는 반면, entire 모드에서는 전체를 암호화한다. 이렇게 일부만 암호화하더라도 파일은 사용 불가능해지면서, 암호화 속도는 크게 빨라진다.



<그림 12> .SINOBI 확장자로 암호화된 파일

Sinobi Ransomware가 사용하는 AES-128-CTR은 카운터 모드 암호화 방식이다. 16바이트의 IV를 카운터로 사용하여 AES 블록 암호로 암호화한 값을 생성하고, 이를 원본 데이터와 XOR하여 암호문을 만든다. 16바이트를 처리할 때마다 카운터를 1씩 증가시키며, AES 블록 암호는 국제 표준을 따르는 10라운드 구현이다.

암호화된 파일의 끝에는 복호화에 필요한 정보를 담은 116바이트의 푸터가 붙는다. 푸터에는 공개키만 저장되고 실제 암호화 키는 저장되지 않는다. 공격자는 자신의 개인키와 푸터의 임시 공개키로 AES 키를 다시 계산할 수 있지만 제3자는 불가능하다.



<그림 13> Sinobi에 의해 암호화된 파일의 푸터 (출처 : eSentire TRU)

디렉토리 암호화

Sinobi는 지정된 디렉토리를 하위 디렉토리까지 재귀적으로 탐색하며 각 폴더에 랜섬노트를 생성한다. 그러나 windows, program files, program files (x86), $RECYCLE.BIN, appdata 폴더는 암호화에서 제외된다. 또한 파일 확장자 기준으로 .exe, .msi, .dll은 제외되고 파일명에 'SINOBI'가 포함되어 있거나 'README.txt'인 파일도 제외하여 이중 암호화와 랜섬노트 손상을 방지한다. 특이사항으로는 .lynx 확장자도 제외 목록에 포함되어 있어, Lynx Ransomware와 Sinobi는 코드 공유 관계라는 점을 보여준다.

바탕화면 변경

모든 파일 암호화가 완료되면, --no-background 및 --no-print 플래그가 지정되지 않은 경우 바탕화면 변경과 프린터 출력을 통해 피해자에게 공격 사실을 알린다.

랜섬노트 텍스트를 화면 해상도에 맞는 비트맵 이미지로 생성하여 %TEMP%₩background-image.jpg 경로에 저장한다. 그 후 레지스트리를 수정하고 SystemParametersInfoW 함수로 배경화면을 변경한다.

<그림 14> Sinobi 바탕화면 (출처 : eSentire TRU)

프린터 출력

EnumPrintersW 함수로 시스템에 연결된 모든 로컬 프린터를 나열한 뒤, PDF 등 가상 프린터는 제외하고 실제 물리 프린터에 대해 랜섬노트를 출력한다. 프린트를 위해 사용하는 함수는 OpenPrinterW -> StartDocPrinterW -> StartPagePrinter -> WritePrinter -> EndPagePrinter -> EndDocPrinter 순서이다. Sinobi는 모든 물리 프린터에서 출력이 완료될 때까지 이 루프를 반복한다.

```
result = OpenPrinterW((LPWSTR)*v4, &phPrinter, 0);
if ( !result )
  goto LABEL_20;
*(_QWORD *)pDocInfo = L"My Document";
v14 = 0;
v15 = L"RAW";
started = StartDocPrinterW(phPrinter, 1u, pDocInfo);
v6 = phPrinter;
if ( started )
{
  if ( StartPagePrinter(phPrinter) )
  {
    sub_140001670(L"[*] Sending note to printer: %s...\n", *v4);
    v8 = lstrlenA(v0);
    if ( !WritePrinter(phPrinter, v0, v8, &pcWritten) )
    {
      EndPagePrinter(phPrinter);
      EndDocPrinter(phPrinter);
      result = ClosePrinter(phPrinter);
      goto LABEL_20;
    }
    v9 = EndPagePrinter(phPrinter);
    v7 = phPrinter;
    if ( v9 )
    {
      v10 = EndDocPrinter(phPrinter);
```

<그림 15> 프린터 출력 로직

4) 결론

Sinobi Ransomware는 INC → Lynx → Sinobi로 이어지는 다세대 랜섬웨어 계보의 최신 진화형으로, 전세대의 검증된 암호화 구현을 계승하면서도 운영 성숙도와 조직 규율 면에서 한 단계 발전한 위협이다. 바이너리 수준에서는 Curve25519 ECDH 키 교환과 AES-128-CTR 대칭 암호화를 결합한 하이브리드 암호화 기법을 사용하며, 파일마다 CSPRNG 기반 임시 키를 생성하고 표준 암호학적 프리미티브를 올바르게 구현하여 공격자의 개인키 없이는 독립적인 복호화 도구 제작이 수학적으로 불가능하다.

특히 주목할 점은 의료 부문에 대한 공격 강화이다. Comparitech의 2025년 의료 랜섬웨어 통계에 따르면, Sinobi는 의료 기관에 대한 랜섬웨어 공격 주장 건수에서 Qilin(66건), INC(45건), SafePay(29건)에 이어 4위(24건)를 기록했다. 2026년 1월 기준 CVMG(Cardiovascular Medical Group of Southern California), Central Jersey Medical Center, Watsonville Community Hospital 등 다수의 의료 기관을 공격한 것으로 확인되며, 2026년에 Sinobi가 의료 부문에 대한 중대한 위협이 될 것으로 전망하고 있다.

Sinobi는 등장 6개월 만에 215건 이상의 피해를 기록하며 2025년 하반기 가장 빠르게 성장한 랜섬웨어 그룹 중 하나로 자리매김했다. INC에서 Lynx로, 다시 Sinobi로 이어지는 코드 재활용과 리브랜딩의 순환은 법 집행 기관의 개입에도 불구하고 랜섬웨어 생태계가 끊임없이 적응하고 진화한다는 사실을 보여준다. 조직은 이러한 위협이 일회성이 아닌 지속적이고 진화하는 위협임을 인식하고, 사전 예방적 보안 태세를 갖추어야 한다.

5) IOCs

SHA256

1b2a1e41a7f65b8d9008aa631f113cef36577e912c13f223ba8834bbefa4bd14
c88f60dbae08519f2f81bb8efa7e6016c6770e66e58d77ab6384069a515e451c
eb3e2a6a50f029fc646e2c3483157ab112f4f017406c3aabedaae0c94e0969f6
f4cd7ab04b1744babef19d147124bfc0e9e90d557408cc2d652d7192df61bda9
e3c080e322862d065649c468d20f620c3670d841c30c3fe5385e37f4f10172e7
e62df17150fcb7fea32ff459ef47cdd452a21269efe9252bde70377fd2717c10
53e2c2d83813d1284ddb8c68b1572b17cca95cfc36a55a7517bf45ff40828be5
43d4847bf237c445ed2e846a106e1f55abefef5c3a8545bd5e4cad20f5deb9a4
4c2429fc8b8ec61da41cbba1b8184ec45fa93a9841b4ca48094bba7741b826b8
694d729d67f1b0c06702490bfab1df3a96fe040fe5d07efa5c92356c329757be
edae3b75deb8013bd48ac4534cca345b90938a2abb91672467c2bf9ae81ff683
0814a0781ab30fca069a085dba201d6fd0f414498fafa4bb42859786d91d4781
59b4deee977e9e27b60e7e179d54a1ce8e56624e73b799523416eee828bfaf76

URL

http://sinobi6ftrg27d6g4sjdt65malds6cfptlnjyw52rskakqjda6uvb7yd.onion
http://sinobi6rlec6f2bgn6rd72xo7hvds4a5ajiu2if4oub2sut7fg3gomqd.onion
http://sinobi6ywgmmvg2gj2yygkb2hxbimaxpqkyk27wti5zjwhfcldhackid.onion
http://sinobi7l3wet3uqn4cagjiessuomv75aw3bvgah4jpj43od7xndb7kad.onion
http://sinobi7sukclb3ygtorysbtrodgdbnrmgbhov45rwzipubbzhiu5jvqd.onion
http://sinobi23i75c3znmqqxxyuzqvhxnjsar7actgvc4nqeuhgcn5yvz3zqd.onion
http://sinobia6mw6ht2wcdjphessyzpy7ph2y4dyqbd74bgobgju4ybytmkqd.onion
http://sinobicrh73ongfuxjajmlyyhalvkhlcgttxkxaxz3gvsgdcgf76uiqd.onion/login

6) YARA

```
rule Sinobi_Ransomware {
    meta:
        description = "Detects Sinobi ransomware binary"
        author = "Piolink"
        date = "2026-02-10"

    strings:
        $magic = "SINOBI" ascii
        $ransom_note = "README.txt" ascii wide
        $extension = ".SINOBI" ascii wide
        $api1 = "SHEmptyRecycleBinA" ascii
        $api2 = "CryptGenRandom" ascii

    condition:
        uint16(0) == 0x5A4D and
        filesize < 5MB and
        $magic and 2 of ($ransom_note, $extension, $api1, $api2)
}
```

```
rule Sinobi_Encrypted_File {
    meta:
        description = "Detects files encrypted by Sinobi ransomware"
        author = "Piolink"
        date = "2026-02-10"

    strings:
        $magic = "SINOBI" ascii
        $decrypt_marker = { 40 42 0F 00 }  // 0xF4240 decryption size marker

    condition:
        // Look for magic identifier near end of file (footer)
        $magic in (filesize-1000..filesize) and
        $decrypt_marker in (filesize-1000..filesize)
}
```