Keep up with Ransomware

Windows, Linux 환경을 모두 노리는 InterLock 랜섬웨어

■ 개요

2024 년 10 월 랜섬웨어 피해 사례는 지난 9 월(406 건)에 비해 약 35% 증가한 550 건을 기록했다. 이는 여러 신규 랜섬웨어 그룹의 등장과 활동을 중단했던 많은 그룹들이 다시 활동을 재개했기 때문이다.

신규 그룹 Sarcoma 는 활동을 시작한 지 한 달 만에 41 건의 공격 사례를 게시하며, 10 월 랜섬웨어 그룹 중 세 번째로 많은 공격 사례를 기록했다. APT73 그룹은 활동 중단을 밝힌 이후 두 달 만에 Bashe 로 리브랜딩하고 20 건의 공격 사례 게시하며 활동을 재개했다.

랜섬웨어로 인한 피해 사례가 꾸준히 늘어나며 국제 수사 기관의 압박이 거세지고 있다. 영국 내무부 산하 법집행기관 NCA 는 LockBit 그룹의 범죄 인프라를 무력화하는 작전인 Cronos Operation 에 대한 추가 소식을 공개했다. 주요 내용은 LockBit 관계자에 대한 신상 공개 및 체포 소식이다. LockBit 계열사로 활동하며 최소 1 억 달러(약 1,380 억 원)를 몸값으로 갈취한 계열사 Beverly 의 주요 정보를 공개했고, 국제 공조를 통해 LockBit 의 개발자와 활동에 참여한 용의자 2 명 및 BPH[1] 서비스를 제공한 관계자를 체포했다. 또한, LockBit 의 인프라 서버 9 개를 압수했다.

압수한 인프라를 분석한 결과, LockBit 그룹은 몸값을 받은 뒤에도 다크웹 유출 사이트에서 게시글만 제거할 뿐 원본 데이터는 삭제하지 않았다. 장기간에 걸친 인프라 무력화 작전으로 인해 LockBit 은 10 월 신규 피해자 게시가 2 건에 그치는 등 활동량이 눈에 띄게 감소했다.

최근 취약점을 악용하는 랜섬웨어 공격이 다수 확인되고 있다. Akira, Fog 랜섬웨어 그룹은 VMware vSphere, Hyper-V 와 같은 가상화 환경의 복구 솔루션 Veeam Backup and Replication 에서 발견된 CVE-2024-40711 취약점을 악용했다. 해당 취약점은 신뢰할 수 없는 데이터나 악성 페이로드[2]로 인해 원격 코드 실행이 가능하다. 패치 이후 취약점 기술 분석과 PoC[3] 코드가 공개돼 랜섬웨어 그룹이 악용한 사례이다.

또한, 웹 호스팅 제어판인 CyberPanel 에서 원격 코드 실행 취약점(CVE-2024-51378[4])이

---

[1] BPH (Bullet Proof Hosting): 법 집행 기관의 요청을 무시하거나 회피하며 웹 호스팅을 제공하는 서비스로, 불법적인 온라인 활동에 주로 사용된다.

[2] 페이로드 (payload): 컴퓨터 시스템에 침투, 변경 또는 기타 방식으로 손상을 입히도록 설계된 코드

[3] PoC (Proof of Concept): 특정 취약점이 실행 가능하다는 것을 증명하는 코드

[4] CVE-2024-51378: 공격자가 인증을 우회하고 임의의 명령을 실행할 수 있는 원격 코드 실행 취약점

발견됐다. PSUAX 랜섬웨어는 해당 취약점을 악용해 시스템의 루트 권한을 얻어 파일을 암호화했고, 발견 당시 약 2 만 대의 서버가 위협에 노출되어 있었다. 다만, 해당 랜섬웨어의 암호화 스크립트는 RSA 개인키가 그대로 노출되어 있기 때문에 공개된 복호화 스크립트를 활용해 별도 비용 지불 없이 암호화된 파일을 복구할 수 있다. 이외에는 .locked(Conti v3 기반), .encryp(Babuk 소스코드 기반) 확장자를 사용하는 2 개의 다른 랜섬웨어와 크립토마이너[5]가 배포되었다.

북한 배후의 위협 그룹 Andariel 과 Play 그룹이 공격에 동일한 계정을 사용한 정황이 발견됐다. 지난 5 월 Andariel 은 손상된 사용자 계정을 악용해 공격 대상에 초기 침투했으며, 오픈소스 C2[6] 프레임워크 Silver 와 Lazarus 그룹이 개발한 원격 관리 도구 DTrack 을 이용해 내부 인프라에 침투해 세션을 유지했다. 9 월에도 초기 침투 때와 동일한 계정으로 다시 접근해 자격 증명을 수집하고, EDR[7]을 비활성화한 뒤 Play 랜섬웨어를 배포한 정황이 발견됐다. 다만, Play 그룹은 RaaS[8] 서비스를 제공하지 않는다고 공식적으로 밝혔기 때문에 Andariel 이 Play 그룹의 계열사로 합류한 것인지, 혹은 IAB[9]의 역할만 제공한 것인지는 불분명하다.

랜섬웨어 위협이 지속되고 있는 가운데 다크웹 및 텔레그램에서 국내 랜섬웨어 사고 사례 2 건이 확인됐다. KillSec 그룹이 국내 부동산 전문 데이터 플랫폼을 공격해 탈취한 데이터를 공개했다. 해당 데이터에는 개인정보, 재학증명서, 사업자 등록증이 포함되어 있었다. 텔레그램에서 활동하는 CyberVolk 그룹은 국내 바이오 연구소의 웹페이지에서 관리자 패널에 접근해 수집한 로그를 판매하는 글을 업로드했다.

---

[5] 크립토마이너: 감염된 PC나 서버의 하드웨어 자원을 이용하여 암호화폐를 채굴하는 악성코드

[6] C2 (Command and Control): 감염된 PC나 서버를 대상으로 통신을 유지하고, 추가적인 명령 전달이나 악성코드 다운로드 등을 수행하는 서버

[7] EDR (Endpoint Detection and Response): 컴퓨터와 모바일, 서버 등 단말기에서 발생하는 악성 행위를 실시간으로 감지하고 분석 및 대응하여 피해 확산을 막는 솔루션

[8] RaaS (Ransomware-as-a-Service): 금전을 대가로 랜섬웨어 코드나 공격에 필요한 도구를 제공하는 비즈니스 모델

[9] IAB (Initial Access Broker): 네트워크 및 시스템의 액세스 권한을 획득한 뒤 금전을 대가로 판매하는 위협 행위자

■ 랜섬웨어 뉴스

infosec

NCA, Cronos Operation 추가 정보 공개

- Evil Corp 소속의 LockBit 계열사 "Beverly"의 신상 공개
- LockBit 개발자, BPH 서비스 관계자 체포 및 LockBit 활동에 연관된 용의자 2명 체포
- 다크웹 유출 사이트와 같은 LockBit 범죄 인프라에 활용한 서버 9개 압수
- 2022년 이후에는 비용을 지불 받아도 탈취한 데이터를 삭제하지 않고 보관하고 있다는 사실 공개

Veeam 백업 솔루션 취약점(CVE-2024-40711)을 악용한 랜섬웨어 그룹

- CVE-2024-40711: 역직렬화로 인해 신뢰할 수 없는 데이터나 악성 페이로드로 원격 코드 실행이 가능한 취약점
- Akira, Fog 랜섬웨어 그룹이 패치되지 않은 서버를 대상으로 공격 수행

PSAUX 랜섬웨어, 웹 호스팅 제어판 CyberPanel의 0-Day 취약점을 악용

- 원격 코드 실행 취약점(CVE-2024-51378)으로 루트 권한 획득 및 파일 암호화 후, 200 달러(한화 약 28만원)의 몸값 요구
- 암호화에 사용한 스크립트(.sh)에 개인키가 저장되어 있어 복호화 가능

KillSec, 국내 부동산 전문 데이터 플랫폼 공격

- 10월 5일, 샘플 데이터와 함께 데이터 공개 협박글 게시
- 개인 정보, 세금 자료, 정부 관련 문서, 사업자 등록증 등이 포함되어 있다고 주장
- 10월 8일, 약 105MB 크기의 데이터 전체 공개

CyberVolk, 국내 바이오 연구소 공격

- 연구소의 홈페이지 관리자 패널에 접근
- 해당 페이지 로그를 판매한다는 글 텔레그램 채널에 업로드

Andariel, Play 랜섬웨어 공격에 연루

- 24년 5월, 공격 대상의 손상된 계정을 활용해 초기 침투 성공
- 초기 침투 이후 Silver, Dtrack 활용해 연결 유지 및 내부 인프라 확산
- 24년 9월, 초기 침투에 사용한 동일 계정으로 자격증명 탈취, 보안 솔루션 비활성화, 랜섬웨어 배포
- Andariel이 IAB 역할만 수행한 것인지, Play 그룹의 계열사로 합류한 것인지 불분명

APT73 그룹 Bashe로 리브랜딩

- 24년 8월 활동을 중단한 APT73 그룹이 Bashe로 리브랜딩 후 활동 재개
- 다크웹 유출 사이트에 기존 피해자 재게시 및 신규 피해자 20건 게시

Apos Security 그룹 활동 재개

- 24년 4월 등장 후, Notion에 피해자를 업로드
- 활동 시작 일주일만에 Notion 페이지 삭제 및 활동을 중단했으나 10월에 다크웹 유출 사이트 신규 개설
- 신규 피해자는 아직 게시되지 않았으며, 기존 피해자만 재게시

Parano Ransomware V1, 텔레그램에서 판매중

- 탐지 우회 기법과 분석 방해 기법이 적용됐으며, 400 달러(한화 약 55만원)에 판매
- 별도의 랜섬웨어 빌더가 존재해 랜섬노트, 배경화면, 암호화 확장자, 암호화폐 지갑 주소 등 변경 가능

Dragon Team, 랜섬웨어 개발 및 RaaS 제공 예고

- 10월 등장한 그룹으로, 텔레그램에서 활동하며 홈페이지 변조, 데이터 탈취, 랜섬웨어 공격 수행
- Dragon Ransomware를 사용하며, 이를 기반으로 RaaS 제공할 것이라고 예고

그림 1. 랜섬웨어 동향

■ 랜섬웨어 위협

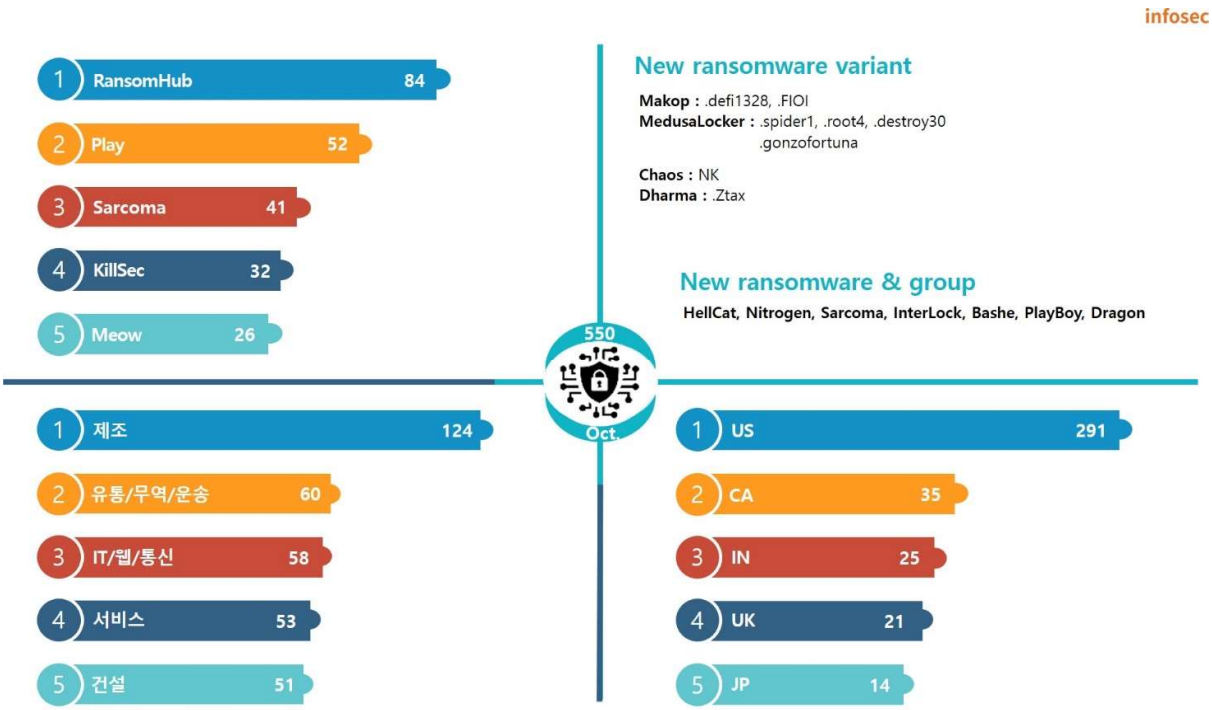


그림 2. 2024 년 10 월 랜섬웨어 위협 현황

새로운 위협

10 월에는 기존에 활동하던 그룹의 리브랜딩 및 활동 재개 소식이 다수 확인됐다. 지난 4 월 등장한 Apos Security 그룹은 Notion 페이지를 통해 피해자를 게시한 이력을 가지고 있는 그룹인데, 활동 일주일 만에 게시글을 삭제한 뒤 행적을 감춘 바 있다. 이후 10 월 다크웹 유출 사이트를 새로 개설해 기존 피해자와 함께 1 건을 추가 게시하며 재개했다. APT73 그룹은 8 월 29 일 이후 활동을 중단한 뒤 10 월에 Bashe 로 그룹명을 변경했으며 신규 피해자 20 건을 업로드하며 재개했다. 이외에도 Nitrogen 그룹 11 건, Sarcoma 그룹 41 건, InterLock 그룹 6 건, HellCat 그룹 1 건, PlayBoy 그룹 1 건을 게시하는 등 신규 랜섬웨어 그룹의 활동도 증가했다.

```
-----BEGIN PGP SIGNED MESSAGE-----
Hash: SHA512

On [22/10/2024], HELLCAT was just an idea. Only two days later, we launched our first attack quick, right?

Now, were taking the HELLCAT servers offline for a few days to get ready for whats next. Weve got targets, and were making sure everythings in place.

Wait for us ... #HELLCAT.

-----BEGIN PGP SIGNATURE-----

iHUEARYKAB0WIQQqAxqbiuUi4RkM//sHznxV9M4/gQUCZx42jwAKCRAHznxV9M4/
gU5sAQCACwLfBEnjdmzdg/hE8WDJncY81HLVG9Lk2ZIRGJIJkQD+KKQFElhPoJ+Y
l1iWVw69RH2V5B31bjelts6WmogNdAQ=
=lfkz
-----END PGP SIGNATURE-----
```

그림 3. HellCat 공지사항

신규 랜섬웨어 그룹 HellCat 그룹은 자신들의 다크웹 유출 사이트에 이스라엘 단원제 입법부인 Knesset 의 내부 문서 45GB 가량을 탈취해 몸값 20 만 달러(약 2 억 8,000 만 원)를

요구했다. 해당 게시물은 약 3 일 만에 삭제됐고, 다음 활동까지 서버를 비활성화 한 뒤 다음 공격에 성공하면 복귀하겠다는 공지를 남기는 등 독특한 모습을 보였다.

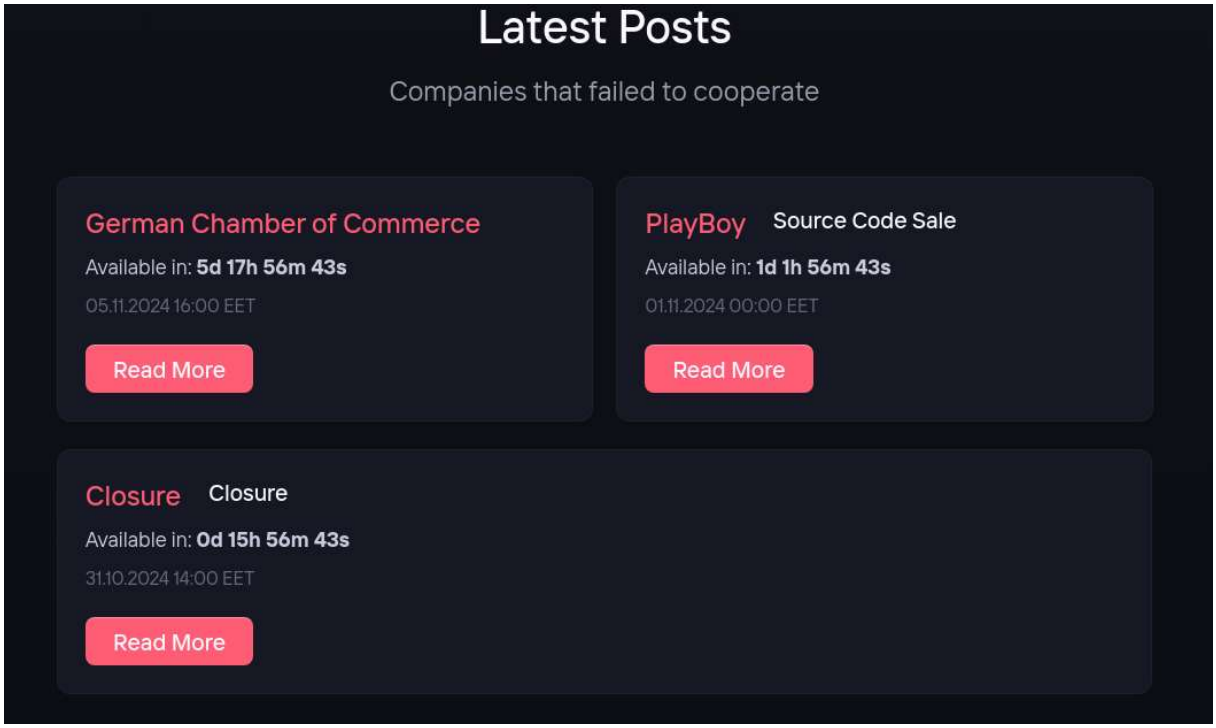


그림 4. PlayBoy 다크웹 유출 사이트

한편, 등장 이틀 만에 활동을 종료한 그룹도 확인됐다. 10 월 28 일 등장한 PlayBoy 그룹은 독일 상공회의소를 공격해 2,800 만 달러 (약 386 억 원)를 요구했다. 하지만, 게시글에는 탈취한 샘플 데이터, 데이터의 종류, 데이터의 크기 등은 기재되어 있지 않았다. 이후 이틀 만에 갑작스러운 사이트 폐쇄를 예고하더니 소스코드와 다크웹 유출 사이트, 관리 패널을 포함해 모든 인프라 판매 글을 게시했다. 31 일 이후로는 다크웹 유출 사이트에 접속할 수 없는 상태다.

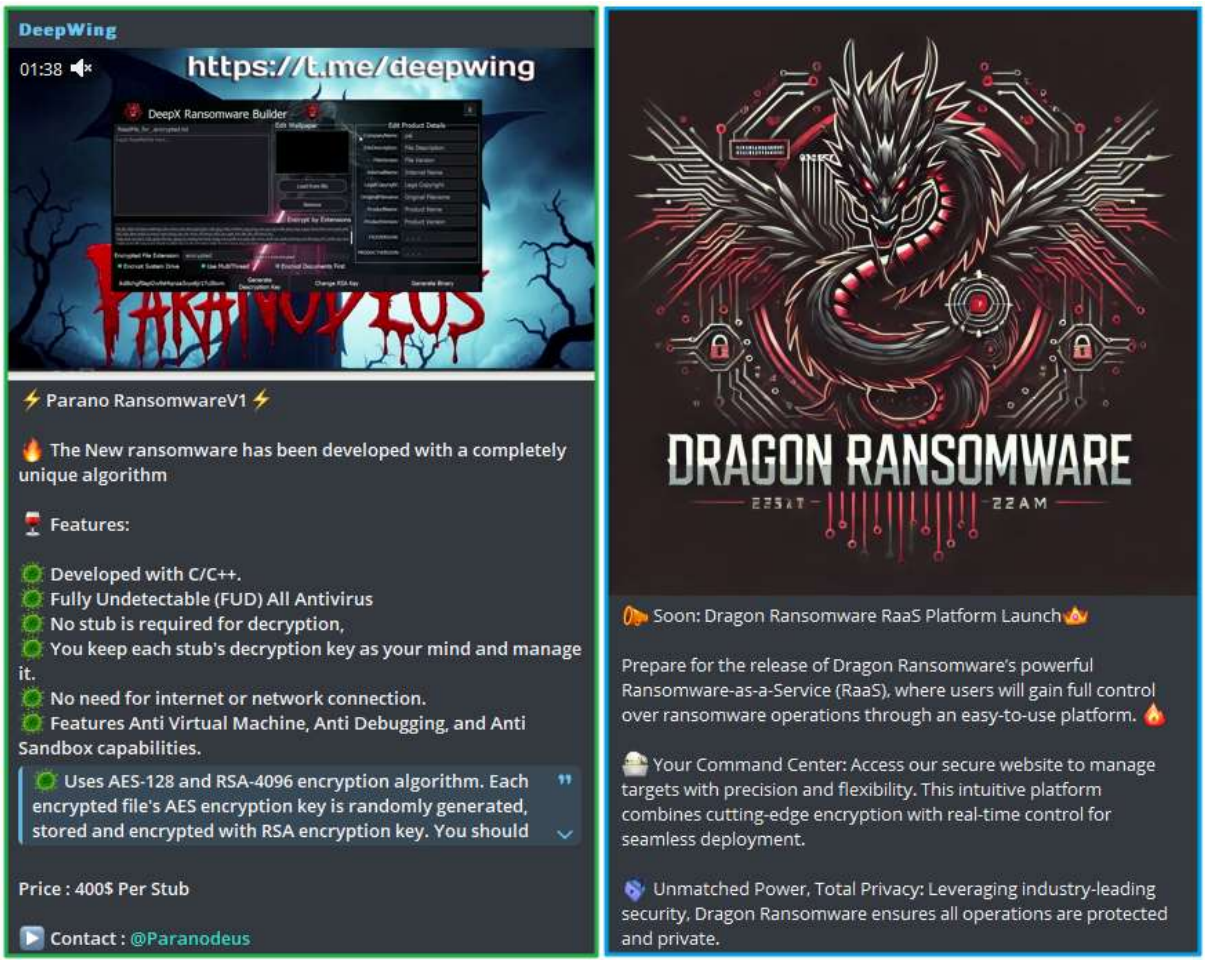


그림 5. 텔레그램 신규 랜섬웨어(좌: Parano Ransomware V1, 우: Dragon Ransomware)

텔레그램을 활용한 랜섬웨어 위협도 지속적으로 확인된다. 10 월 19 일 각종 탐지 회피 기법이 적용된 Parano 랜섬웨어를 400 달러(약 55 만 원)에 판매하는 글이 한 채널에 업로드 됐다. 또한, 텔레그램에서 활동하는 신규 Dragon 랜섬웨어 그룹이 등장했다. 자신들의 랜섬웨어를 이용해 공격한 뒤 탈취한 데이터를 텔레그램에서 판매 중인데, 최근에는 랜섬웨어를 서비스 형태로 제공하는 Dragon RaaS 를 소개했다. 아직 랜섬웨어만 개발된 단계로 서비스를 제공하지는 않지만, 추후 랜섬웨어와 관리 페이지까지 포함된 RaaS 서비스를 제공할 것이라고 밝혔다.

Top5 랜섬웨어

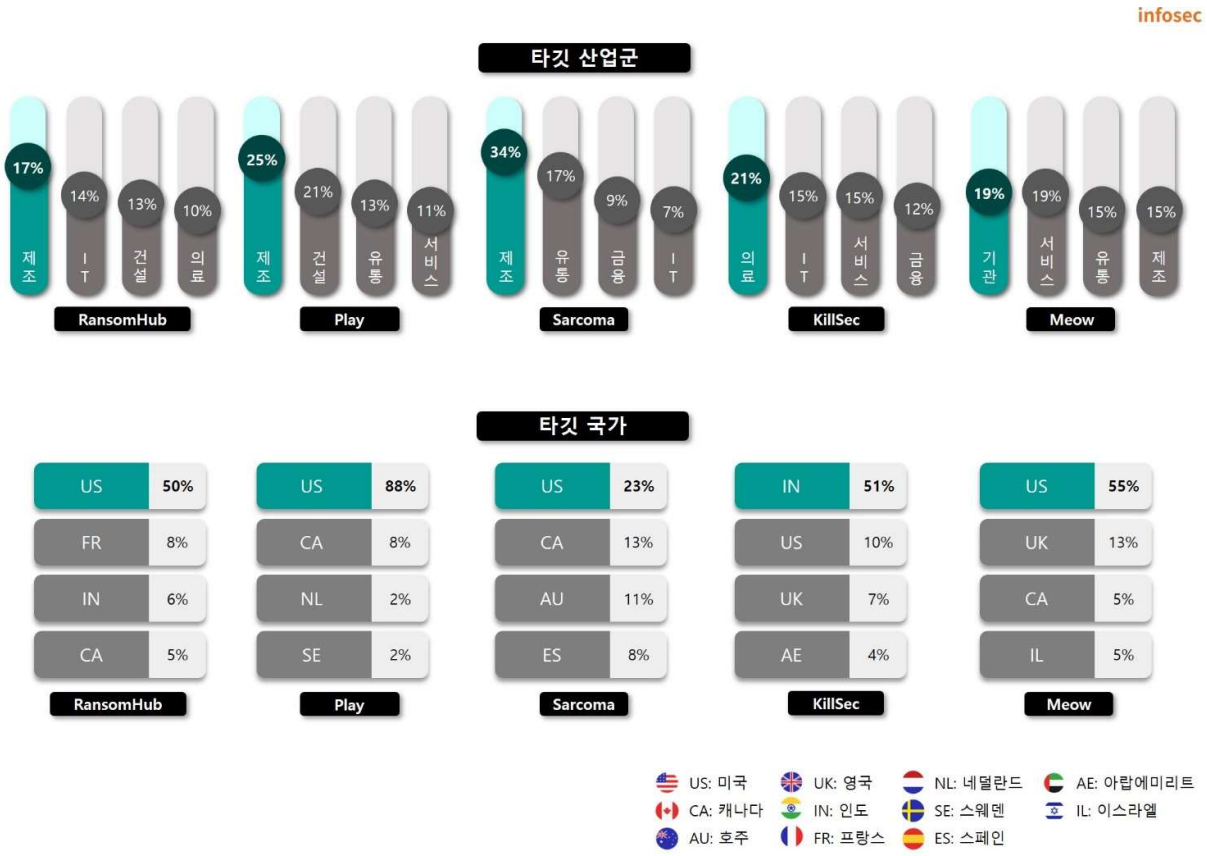


그림 6. 산업/국가별 주요 랜섬웨어 공격 현황

RansomHub 그룹은 10 월에만 84 건의 피해자를 게시하며 위협적인 그룹으로 자리매김했다. 최근 멕시코 일부 지역의 공항을 감독하는 Grupo Aeroportuario del Centro Norte(OMA)를 공격해 회계 및 투자 자료, 고객 개인 정보, 자격 증명 및 비밀번호, 데이터베이스 등 3TB 에 달하는 데이터를 탈취했다. 게다가, 이번 공격으로 OMA 가 관리하는 공항은 시스템 마비로 인해 공항 스크린이 작동하지 않는 등 업무에도 큰 차질이 생겼다.

Play 그룹은 미국의 광대역 서비스 업체 OzarksGo 를 공격해 서비스를 마비시킨 뒤 개인 정보, 기밀 문서, 고객 문서, 예산 정보, 급여 및 계약서가 포함된 데이터를 탈취했다. 이로 인해 OzarksGO 를 이용하는 고객들은 10 월 7 일부터 TV 서비스를 원활하게 이용하지 못하고 있다. 기업의 공식 성명에 따르면 서비스 장애는 장기간 지속될 예정으로, TV 서비스 요금 면제, 기존 TV 서비스를 스트리밍 TV 서비스로 무료 전환 등의 조치를 시행하고 있다. 랜섬웨어 공격으로 인한 서비스 장애와 이에 따른 후속 조치로 상당한 손실을 입고 있다.

Sarcoma 그룹은 10 월에 새로 등장한 그룹으로 한 달 만에 41 건의 피해자를 게시했다. 기업 및 사업체에 보험을 제공하는 업체 Ferrer&Ojeda 를 공격해 계약서, 직원 개인 정보, 주요 DB 정보가 포함된 1.27TB 크기의 데이터를 탈취해 공개했다.

KillSec 그룹은 10 월에만 피해자를 32 건 게시했는데 활동 이래 가장 많은 피해자 수다. 또한, UI 를 개선한 새로운 다크웹 유출 사이트 KillSecurity 3.0 을 공개했다. 지난 10 월에는 국내

부동산 전문 데이터 플랫폼을 공격해 개인정보, 재학증명서, 사업자 등록증 등의 데이터를 탈취해 공개했다.

Meow 그룹은 이스라엘 보안 회사인 Modiin Ezrachi 를 공격했다. 해당 기업은 이스라엘 점령지나 정착촌에 보안 및 경비 서비스를 제공하고, 이스라엘 정부와 계약을 맺어 교육 기관 및 정부 시설을 보호하는 역할뿐만 아니라 서안 지구의 주요 검문소를 운영하는 기업이다. Meow 그룹은 Moddin Ezrachi 의 직원 정보, 정부 계약서, 보안 패스와 같은 486GB 의 민감한 데이터를 탈취했다.

■ 랜섬웨어 집중 포커스

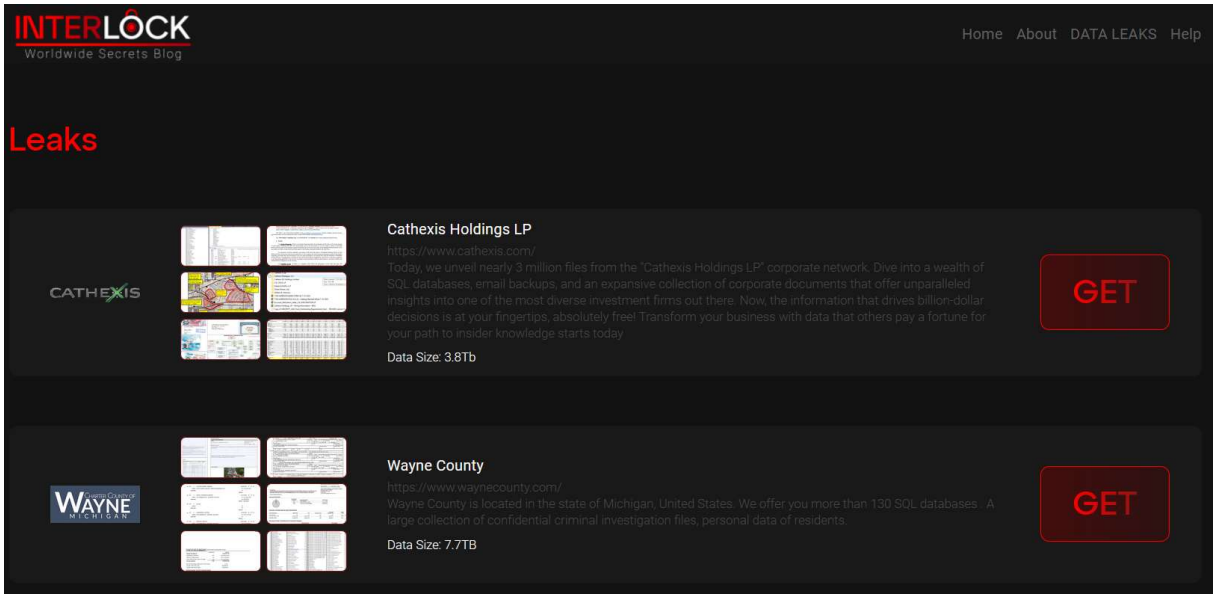


그림 7. InterLock 다크웹 유출 사이트

InterLock 그룹은 지난 10 월 9 일 새로 발견된 그룹이다. 발견 당시 다크웹 유출 사이트에는 별도의 피해자가 게시되지 않은 상태였지만, 13 일부터 피해자를 게시하기 시작했다. 이들은 공격 이후 파일 복호화와 유출 데이터 공개 방지를 위해 총 4 일의 시간을 제공한다. 시간 내에 비용을 지불하면 복호화와 함께 탈취한 데이터를 파기하지만, 협상 기간이 지나거나 협상이 제대로 이루어지지 않으면 복호화 키를 파괴한 뒤 데이터를 판매하거나 공개한다고 협박한다.

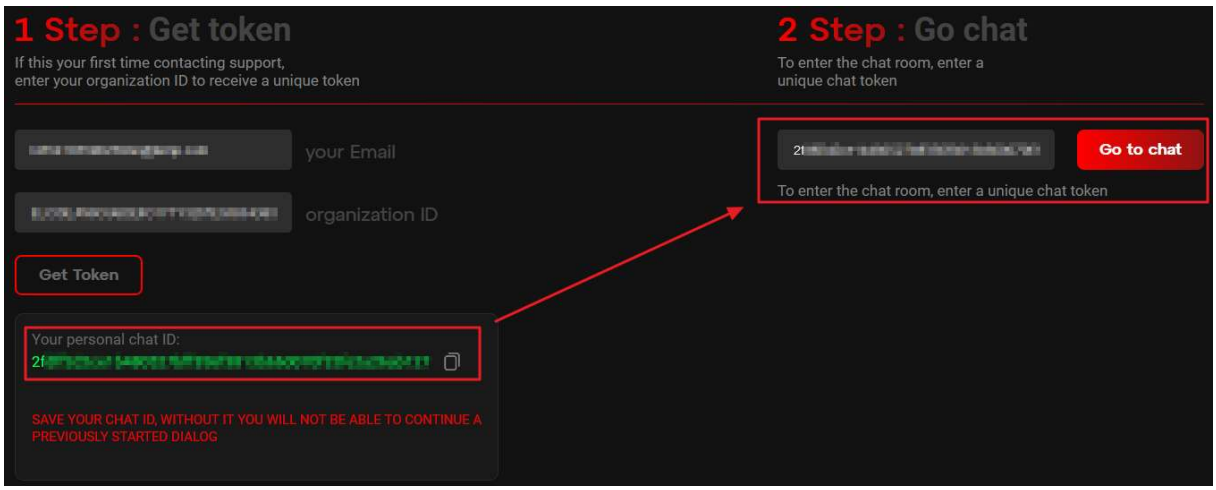


그림 8. InterLock 다크웹 협상 페이지

InterLock 그룹은 랜섬노트를 통해 피해자가 채팅 페이지에 접근할 수 있도록 유도한다. 페이지 주소와 접근 방법을 설명한 뒤 랜섬노트에 기재된 ID 와 사용자 이메일 주소를 입력하면 고유한 채팅방을 만들 수 있는 토큰을 제공한다. 이후 피해자는 채팅방에서 공격자와 협상을 할 수 있다.

```
initRand();
params(argc, argv);
if ( systemArg )
  return addScheduledTask(argc, argv);
ThreadInit();
if ( pathFile )
  threadStart(&pathFile);
if ( pathDir )
  loopdir(&pathDir);
if ( !pathDir && !pathFile )
  allloop();
sleep(1);
waitThread();
threadFree();
sleep(2);
if ( delArg )
  deleteme();
jEvtClearLog();
return 0;
```

```
initRand();
params(argc, argv);
threadInit();
if ( pathFile )
  threadStart(&pathFile);
if ( pathDir )
  loopdir(&pathDir);
if ( !pathDir && !pathFile )
  allLoop();
sleep(1u);
waitThread();
threadFree();
sleep(2u);
if ( (del & 1) != 0 )
  removeme(*argv);
return 0;
```

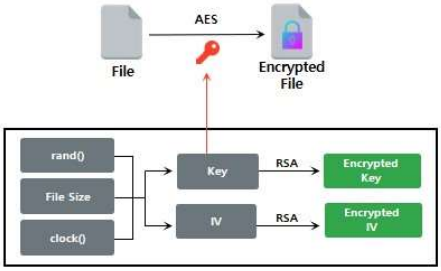
그림 9. InterLock 랜섬웨어 코드 비교(좌: Windows, 우: Linux)

InterLock 랜섬웨어는 Windows 버전과 Linux 버전이 존재한다. 두 버전은 인자를 확인하고 멀티스레드를 기반으로 파일을 암호화하는 부분에서는 거의 동일하게 동작한다. 다만, OS 차이에 따른 일부 모듈이나 함수, 예외 디렉터리 및 파일에는 차이가 있다. 이외에도 Windows 버전에는 랜섬웨어를 작업 스케줄러에 등록하고, 파일 암호화 이후 이벤트 로그를 삭제하는 것처럼 기능적으로도 차이가 존재한다. 따라서, 이번 보고서에서는 두 버전 간의 유사점과 차이점을 분석하여 각 운영체제에 따른 동작 방식을 상세하게 다루고자 한다.

InterLock Ransomware

암호화 키

AES 알고리즘으로 파일 암호화를 하고 RSA 알고리즘으로 키와 IV를 보호



암호화 방식

1MB 크기의 블록 단위로 패딩 후 파일 전체 암호화

특징

랜섬노트

!_README_!.txt - Windows 메모장

파일(F) 편집(E) 서식(O) 보기(V) 도움말(H)

INTERLOCK - CRITICAL SECURITY ALERT

To Whom It May Concern,
Your organization has experienced a serious security breach. Immediate action is required to mitigate further risks.

THE CURRENT SITUATION
- Your systems have been infiltrated by unauthorized entities.
- Key files have been encrypted and are now inaccessible to you.
- Sensitive data has been extracted and is in our possession.

WHAT YOU NEED TO DO NOW
1. Contact us via our secure, anonymous platform listed below.
2. Follow all instructions to recover your encrypted data.

Access Point: http://ebhmkoohccl45qesdbvrjqtyro2hmhkmh6vkyfyjjzfllm3ix72aqaid.onion/support/step.php
Use your unique Company ID: [redacted]

DO NOT ATTEMPT:
- File alterations: Renaming, moving, or tampering with files will lead to irreversible damage.
- Third-party software: Using any recovery tools will corrupt the encryption keys, making recovery impossible.
- Reboots or shutdowns: System restarts may cause key damage. Proceed at your own risk.

!_README_!.txt

변경 확장자

.interlock

제작 언어

C/C++

그림 10. InterLock 랜섬웨어 개요

InterLock 랜섬웨어 전략

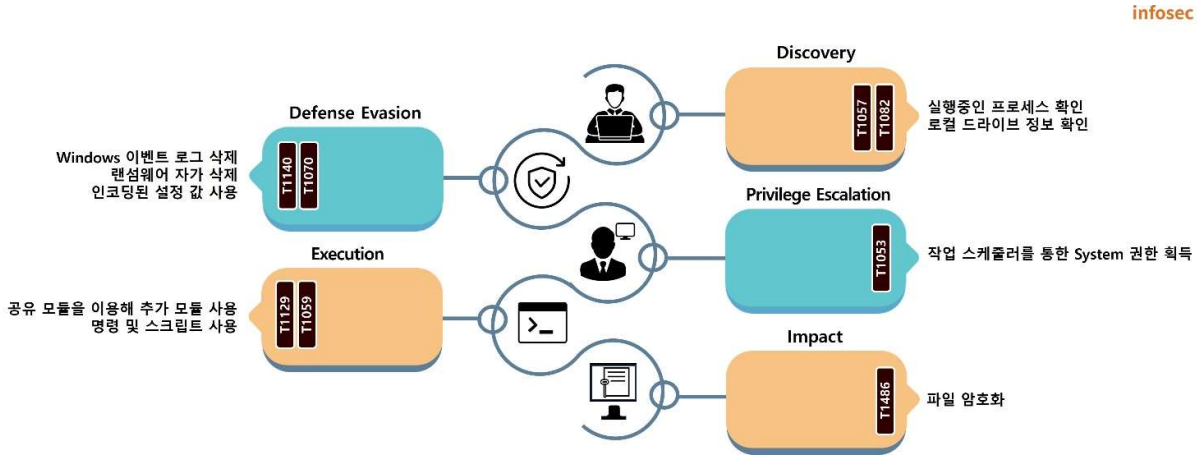
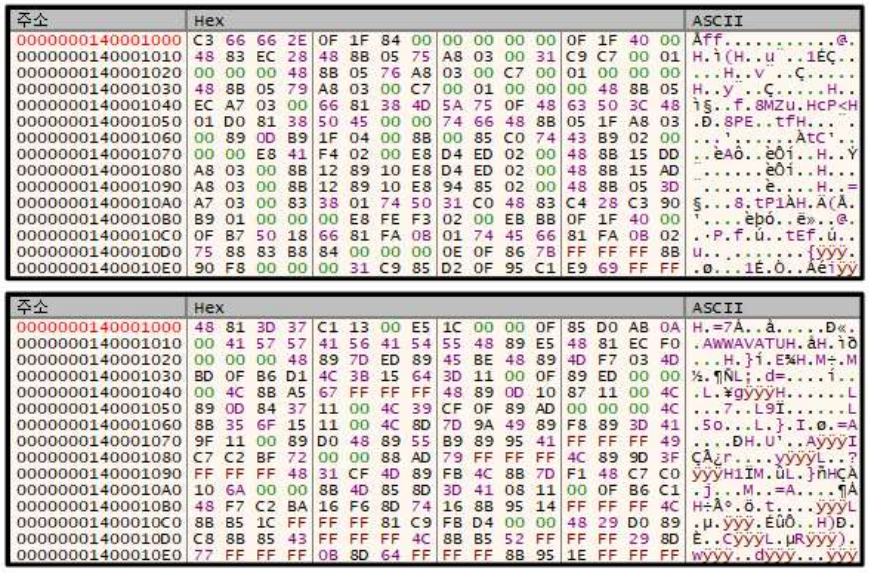


그림 11. InterLock 랜섬웨어 공격 전략

Linux 버전의 랜섬웨어는 바로 실행 인자를 확인하는 반면, Windows 버전은 실행 가능한 원본 코드로 복구한 뒤 실행하는 코드 패치 기법을 사용한다. 이에 따라 원본 코드를 복구하는 과정이 우선적으로 진행된다.



주소	Hex	ASCII
0000000140001000	C3 66 66 2E 0F 1F 84 00 00 00 00 00 0F 1F 40 00	Åff.....@.
0000000140001010	48 83 EC 28 48 8B 05 75 A8 03 00 31 C9 C7 00 01	H.ì(H..u¨..1ÉÇ..
0000000140001020	00 00 00 48 8B 05 76 A8 03 00 C7 00 01 00 00 00	...H..v¨..Ç.....
0000000140001030	48 8B 05 79 A8 03 00 C7 00 01 00 00 00 48 8B 05	H..y¨..Ç.....H..
0000000140001040	EC A7 03 00 66 81 38 4D 5A 75 0F 48 63 50 3C 48	ì§..f.8MZu.HcP<H
0000000140001050	01 D0 81 38 50 45 00 00 74 66 48 8B 05 1F A8 03	.Ð.8PE..tfH...¨.
0000000140001060	00 89 0D B9 1F 04 00 8B 00 85 C0 74 43 B9 02 00	...'.....ÀtC'..
0000000140001070	00 00 E8 41 F4 02 00 E8 D4 ED 02 00 48 8B 15 DD	..èAô..èÔí..H..Ý
0000000140001080	A8 03 00 8B 12 89 10 E8 D4 ED 02 00 48 8B 15 AD	¨.....èÔí..H...-
0000000140001090	A8 03 00 8B 12 89 10 E8 94 85 02 00 48 8B 05 3D	¨.....è....H..=
00000001400010A0	A7 03 00 83 38 01 74 50 31 C0 48 83 C4 28 C3 90	§...8.tP1ÀH.Ä(Ã.
00000001400010B0	B9 01 00 00 00 E8 FE F3 02 00 E8 BB 0F 1F 40 00	'....èþó..è»..@.
00000001400010C0	0F B7 50 18 66 81 FA 0B 01 74 45 66 81 FA 0B 02	.·P.f.ú..tEf.ú..
00000001400010D0	75 88 83 B8 84 00 00 00 0E 0F 86 7B FF FF FF 8B	u..¸.....†{ÿÿÿ.
00000001400010E0	90 F8 00 00 00 31 C9 85 D2 0F 95 C1 E9 69 FF FF	.ø...1É.Ò..Áéiÿÿ

주소	Hex	ASCII
0000000140001000	48 81 3D 37 C1 13 00 E5 1C 00 00 0F 85 D0 AB 0A	H.=7Á..å.....Ð«.
0000000140001010	00 41 57 57 41 56 41 54 55 48 89 E5 48 81 EC F0	.AWWAVATUH.åH.ìð
0000000140001020	00 00 00 48 89 7D ED 89 45 BE 48 89 4D F7 03 4D	...H.}í.E¾H.M÷.M
0000000140001030	BD 0F B6 D1 4C 3B 15 64 3D 11 00 0F 89 ED 00 00	½.¶ÑL;.d=....í..
0000000140001040	00 4C 8B A5 67 FF FF FF 48 89 0D 10 87 11 00 4C	.L.¥gÿÿÿH......L
0000000140001050	89 0D 84 37 11 00 4C 39 CF 0F 89 AD 00 00 00 4C	...7..L9Ï.....L
0000000140001060	8B 35 6F 15 11 00 4C 8D 7D 9A 49 89 F8 89 3D 41	.5o...L.}.I.ø.=A
0000000140001070	9F 11 00 89 D0 48 89 55 B9 89 95 41 FF FF FF 49	....ÐH.U'..Aÿÿÿ I
0000000140001080	C7 C2 BF 72 00 00 88 AD 79 FF FF FF 4C 89 9D 3F	ÇÂ¿r....yÿÿÿL..?
0000000140001090	FF FF FF 48 31 CF 4D 89 FB 4C 8B 7D F1 48 C7 C0	ÿÿÿH1ÏM.ûL.}ñHÇÀ
00000001400010A0	10 6A 00 00 8B 4D 85 8D 3D 41 08 11 00 0F B6 C1	.j...M..=A....¶Á
00000001400010B0	48 F7 C2 BA 16 F6 8D 74 16 8B 95 14 FF FF FF 4C	H÷Âº.ö.t....ÿÿÿL
00000001400010C0	8B B5 1C FF FF FF 81 C9 FB D4 00 00 48 29 D0 89	.µ.ÿÿÿ.ÉûÔ..H)Ð.
00000001400010D0	C8 8B 85 43 FF FF FF 4C 8B B5 52 FF FF FF 29 8D	È..CÿÿÿL.µRÿÿÿ).
00000001400010E0	77 FF FF FF 0B 8D 64 FF FF FF 8B 95 1E FF FF FF	wÿÿÿ..dÿÿÿ...ÿÿÿ

그림 12. 코드 패치 전후 메모리 비교(상: 코드 패치 전, 하: 코드 패치 후)

실행되는 코드가 저장된 영역에서 동일한 부분을 확인해보면 코드 패치 이후 저장된 데이터, 즉 코드가 변경된 것을 확인할 수 있다. Windows 버전의 InterLock 랜섬웨어는 백신과 같은 보안 프로그램에 탐지되지 않기 위해 랜섬웨어가 실행되면 실행 가능한 원본 코드로 복구한 뒤 실행하는 기법을 사용한다.

Windows 버전과 Linux 버전은 우선적으로 실행 인자를 확인한 뒤 특정 동작의 수행 여부를 결정한다. 두 버전 모두 확인하는 인자는 네 종류로 동일하다. 지정된 디렉터리나 파일만 암호화하는 인자, 실행 이후 랜섬웨어 파일을 스스로 삭제하는 인자가 존재한다. "-s" 인자의 경우 두 버전 모두 입력 여부는 확인하지만, Linux 버전은 확인만 할 뿐 기능이 추가되거나 제거되지 않고, Windows 버전만 스케줄러에 랜섬웨어를 등록하는 기능이 추가된다. 각 실행 인자와 기능은 아래 표와 같다.

인자	설명
--directory [target]	지정한 디렉터리만 암호화
--file [target]	지정한 파일만 암호화
--delete	파일 암호화 이후 자가 삭제
--system	작업 스케줄러 등록 및 권한 상승(Windows)

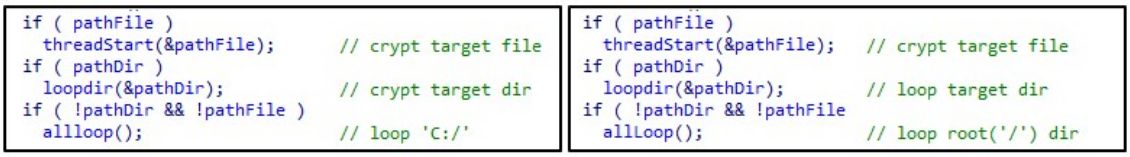
표 1. 실행 인자

Windows 버전은 총 4 개의 작업 스케줄러 명령어를 사용한다. 우선 작업을 등록하기 위해 기존에 존재하는 작업을 삭제하고, 랜섬웨어 실행 명령어에서 --system 인자를 제거한다. 인자를 제거한 명령어를 매일 20 시에 System 권한으로 실행하도록 작업을 등록한 다음 해당 작업을 즉시 실행한 뒤 삭제한다. 작업 스케줄러는 보통 지속성 확보나 권한 상승을 위해 사용되는데, InterLock 랜섬웨어의 경우 System 권한으로 작업 실행 후 작업을 바로 삭제하기 때문에 권한 상승을 위해 사용한 것으로 확인된다. 사용하는 명령어와 설명은 아래 표와 같다.

명령어	설명
schtasks /delete /tn TaskSystem /f > nul	기존 작업 삭제
schtasks /create /sc DAILY /tn "TaskSystem" /tr "cmd /C cd {path} && {execute_command}" /st 20:00 /ru system > nul	랜섬웨어 작업 등록 (System 권한)
schtasks /run /tn TaskSystem > nul	TaskSystem 작업 실행
schtasks /delete /tn TaskSystem /f > nul	TaskSystem 작업 삭제

표 2. 작업 스케줄러 명령어

다음으로는 입력한 인자에 따라 암호화 대상을 설정한 뒤 멀티스레드를 기반으로 파일을 암호화한다. 두 버전 모두 --file 인자를 사용하면 해당 파일만 암호화를 진행하고, --directory 인자를 사용하면 해당 디렉터리와 그 하위에 존재하는 모든 파일을 암호화한다. 두 인자 모두 사용하지 않았을 때는 최상위 디렉터리부터 모두 암호화를 진행하는데, Windows 버전은 C 드라이브의 최상위 디렉터리부터 암호화하고 Linux 버전은 루트 디렉터리부터 암호화를 진행한다.

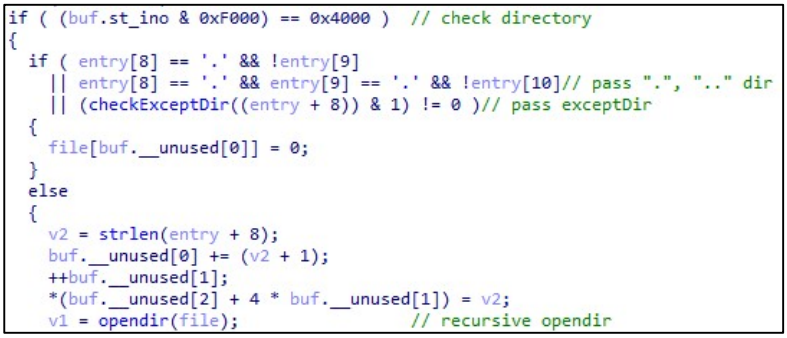


```
if ( pathFile )
  threadStart(&pathFile);       // crypt target file
if ( pathDir )
  loopdir(&pathDir);            // crypt target dir
if ( !pathDir && !pathFile )
  allloop();                    // loop 'C:/'
```

```
if ( pathFile )
  threadStart(&pathFile);    // crypt target file
if ( pathDir )
  loopdir(&pathDir);         // loop target dir
if ( !pathDir && !pathFile
  allLoop();                 // loop root('/') dir
```

그림 13. 실행 인자에 따른 암호화 대상 설정(좌: Windows, 우: Linux)

--directory 인자를 사용해서 특정 디렉터리를 암호화하거나 --directory 인자와 --file 인자를 모두 사용하지 않아 최상위 디렉터리부터 암호화하는 경우, 디렉터리에 존재하는 모든 파일 및 디렉터리를 구분한다. 파일인 경우 암호화 스레드를 호출해 암호화를 진행, 디렉터리인 경우 랜섬노트를 생성하고 그 디렉터리 내부를 재귀적으로 탐색한다.



```
if ( (buf.st_ino & 0xF000) == 0x4000 )  // check directory
{
  if ( entry[8] == '.' && !entry[9]
    || entry[8] == '.' && entry[9] == '.' && !entry[10]// pass ".", ".." dir
    || (checkExceptDir((entry + 8)) & 1) != 0 )// pass exceptDir
  {
    file[buf.__unused[0]] = 0;
  }
  else
  {
    v2 = strlen(entry + 8);
    buf.__unused[0] += (v2 + 1);
    ++buf.__unused[1];
    *(buf.__unused[2] + 4 * buf.__unused[1]) = v2;
    v1 = opendir(file);                 // recursive opendir
```

그림 14. 디렉터리 검증 및 재귀적 탐색

모든 디렉터리를 탐색하는 것은 아니며, 예외 디렉터리는 암호화를 진행하지 않는다. 버전별 확인된 암호화 예외 디렉터리는 아래 표와 같다.

Windows	Linux
.(현재 폴더), ..(상위 폴더), $Recyble.Bin, Boot, Documents and Settings, PerfLogs, ProgramData, Recovery, System Volume Information, Windows, AppData, WindowsApps, Windows Defender, WindowsPowerShell, Windows Defender Advanced Threat Protection	.(현재 폴더), ..(상위 폴더), bin, boot, cdrom, dev, etc, home, lib, lib32, lib64, libx32, lost+found, media, mnt, opt, proc, run, root, sbin, snap, srv, sys, tmp, usr, var

표 3. 암호화 예외 디렉터리

또한, 식별된 대상이 파일인 경우, 해당 파일을 암호화하기 전에 별도의 예외 리스트를 기반으로 암호화 여부를 결정한다. 버전별 확인된 예외 파일 및 확장자는 아래 표와 같다.

Windows	Linux
!__README__!.txt, .bat, .bin, .cab, .cmd, .com, .cur, .diagcab, .diagcfg, .diagpkg, .drv, .hlp, .hta, .ico, .msi, .ocx, .psm1,.scr, .sys, .ini, .url, .dll, .exe, .ps1	!__README__!.txt, . boot.cfg, .sf, .b00, .v00, .v01, .v02, .v03, .v04, .v05, .v06, .v07, t00

표 4. 암호화 예외 파일 및 확장자

우선 파일명에 .interlock 확장자가 존재하는지 확인해 암호화 여부를 파악하고, 암호화되지 않은 파일이면 파일명에 .interlock 확장자를 추가한다. 시스템 시간을 기반으로 랜덤한 AES 키와 초기화 벡터(IV)를 생성한다. 이렇게 생성한 AES 키와 IV 는 파일 암호화에 사용되고, 하드코딩된 RSA 공개키를 이용해 암호화한 뒤 원본 파일의 맨 끝에 저장한다.

```
if ( !checkFileExt(target_file, ".interlock") )
{
  strcat(strcpy(encrypted_fileName, target_file), ".interlock");
  if ( !rename(target_file, encrypted_fileName) )
  {
    Stream = fopen(encrypted_fileName, "rb+");
    if ( Stream )
    {
      file_size = fsize(Stream);
      key_len = 48;
      key_IV = malloc(0x40ui64);
      generateKey(key_IV, file_size, key_len);// generate random key(32Bytes) & IV(16Bytes)
      file_size = addPaddingFile(Stream, file_size);
      *&ElementCount[1] = malloc(0x500ui64);
      *&ElementCount[1] = rsaCrypt(key_IV, key_len, *&ElementCount[1], ElementCount);// encrypt aes key & IV via RSA
```

그림 15. 암호화 여부 확인 및 암호화 키 생성

파일 암호화에는 AES 알고리즘을 사용하며 생성한 키와 IV 를 활용해 CBC 모드로 암호화를 진행한다. 파일 암호화는 1MB 크기의 블록 단위로 진행하며, 파일 전체를 암호화한다.

```
v5 = find_cipher("aes");
if ( cbc_start(v5, a3, a2, 32, 0, v9) )
{
  free(Block);
  free(v9);
  free(Buffer);
}
else
{
  while ( v17 > 0 )
  {
    v6 = v17;
    if ( v17 > ElementCount )
      v6 = ElementCount;
    v8 = fread(Block, 1ui64, v6, a1);
    if ( cbc_setiv(a3, 0x10ui64) || cbc_encrypt(Block, Buffer, v8, v9) )
      break;
    adjustFilePosition(a1, -v8, 1);
    fwrite(Buffer, 1ui64, v8, a1);
```

그림 16. 파일 암호화

--del 인자를 사용했다면 파일 암호화가 끝난 뒤 자가 삭제를 수행해 흔적을 지운다. Linux 버전은 특정 경로를 삭제하는 rmdir 명령어를 활용해 랜섬웨어를 삭제하고, Windows 버전의 경우 랜섬웨어 파일을 삭제하는 DLL 파일을 생성한 뒤, 이를 활용해 자가 삭제를 진행한다.

```
if ( !GetModuleFileNameA(0i64, ransomware, 0x104u) )
  return 0i64;
rand_num = rand();
tmp_path = getenv("tmp");
formatString2(self_deletefile, "%s/tmp%d.wasd", tmp_path, rand_num);
Stream = fopen(self_deletefile, "wb");        // create "%tmp%/tmp{rand_num}.wasd"
if ( !Stream )
  return 0i64;
fwrite(&data, 1ui64, 0xA00ui64, Stream);
fclose(Stream);
formatString2(v4, "rundll32.exe %s,run %s", self_deletefile, ransomware);
return create_process(v4);
```

그림 17. 자가 삭제용 DLL 생성(Windows)

DLL 파일은 랜섬웨어에 하드코딩된 상태로 저장되어 있으며, 이를 임시 폴더에 저장한다. 저장된 DLL 파일은 remove API 를 이용해 인자로 전달된 경로의 파일을 삭제하는 기능만이 포함된 단순 파일이다. Windows 버전은 해당 DLL 파일을 활용해 랜섬웨어를 삭제한다.

```
int __fastcall run(__int64 a1, __int64 a2, const char *a3)
{
  return remove(a3);
}
```

그림 18. tmp.wasd 기능

이외에도 Windows 버전에는 이벤트 로그를 삭제하는 기능이 존재한다. API 를 활용해 Application, Security, System, Forwarded Events 까지 총 4 개의 항목을 모두 삭제한다.

```
EvtClearLog(0i64, L"Application", 0i64, 0);
EvtClearLog(0i64, L"Security", 0i64, 0);
EvtClearLog(0i64, "S", 0i64, 0);
EvtClearLog(0i64, &ystem, 0i64, 0);
return EvtClearLog(0i64, L"Forwarded Events", 0i64, 0);
```

그림 19. 이벤트 로그 삭제

InterLock 랜섬웨어 대응방안

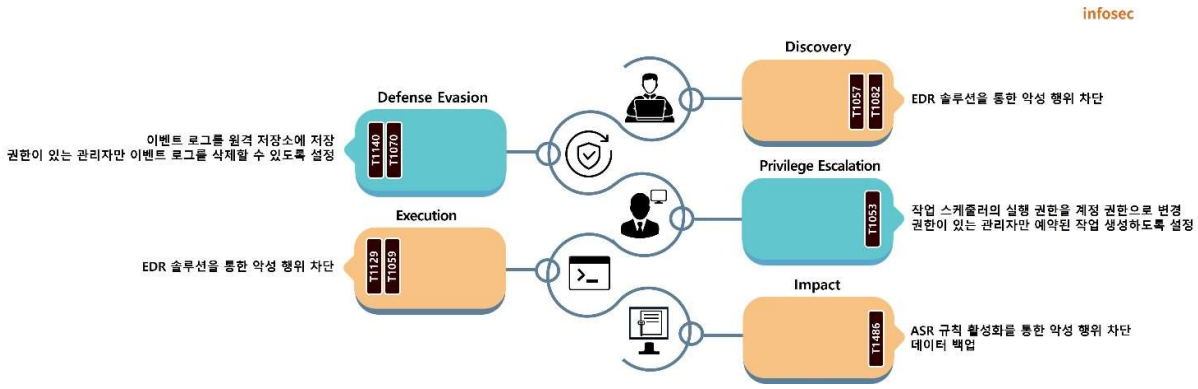


그림 20. InterLock 랜섬웨어 대응방안

InterLock 랜섬웨어의 Windows 버전은 실행 가능한 원본 코드로 복구하는 코드 패치 기법을 사용하기 때문에 Anti-Virus 등의 솔루션으로는 탐지되지 않을 수 있다. 따라서, 행위 기반으로 악성 행위를 식별해 차단하는 EDR 솔루션을 사용해 위협을 차단할 수 있다. 뿐만 아니라 침해 사고 분석에 어려움을 주기 위해 이벤트 로그를 삭제하는데, 이벤트 로그를 원격 저장소에 저장하거나 권한이 있는 관리자만 삭제할 수 있도록 설정해 흔적을 지우지 못하게 할 수 있다.

Windows 버전의 경우 작업 스케줄러를 이용해 System 권한으로 권한 상승을 시도한다. 따라서, 작업 스케줄러를 통해 실행되는 프로세스가 System 권한이 아니라 작업을 생성한 계정의 권한으로 실행되도록 설정, 혹은 관리자 권한이 있는 사용자가 아니면 작업을 등록하지 못하도록 설정하는 등의 조치가 필요하다.

파일 암호화는 물론, 자가 삭제를 위한 프로세스 생성 등을 방지하기 위해 ASR[10] 규칙을 활성화하거나 EDR 솔루션을 통해 공격자가 사용하는 특정 프로세스를 차단해 악성 행위를 막을 수 있다. InterLock 랜섬웨어는 파일 암호화 기능만 존재할 뿐 백업 복사본을 별도로 삭제하지 않기 때문에 Windows 의 기본 기능으로 만든 시스템 백업을 통해 일부 파일을 복구할 수 있다. 이외에도 주요 데이터를 별도의 네트워크나 저장소에 소산 백업해 피해를 최소화할 수 있다.

Linux 버전의 경우 파일 시스템 탐색 후 파일 암호화 및 암호화 이후 자가 삭제하는 기능만 존재한다. 따라서, 랜섬웨어가 실행되더라도 주요 파일을 암호화하지 못하게 사용자 계정의 파일 및 폴더의 권한을 최소한으로 부여해 피해를 최소화할 수 있다. 이외에도 EDR 솔루션을 사용해 악성 프로세스의 실행을 차단하거나 애플리케이션 허용 목록을 설정해 사전에 허용된 프로그램만 실행될 수 있도록 제한할 수 있다. 또한, 데이터를 별도의 네트워크나 저장소에 소산 백업해 피해를 최소화할 수 있다.

[10] ASR (Attack Surface Reduction): 공격자가 사용하는 특정 프로세스와 실행 가능한 프로세스를 차단하는 보호 기능

Indicator Of Compromise

InterLock(Windows)

a26f0a2da63a838161a7d335aaa5e4b314a232acc15dcabdb6f6dbec63cda642

InterLock(Linux)

e86bb8361c436be94b0901e5b39db9b6666134f23cce1e5581421c2981405cb1

28c3c50d115d2b8ffc7ba0a8de9572fbe307907aaae3a486aabd8c0266e9426f

File Name(Windows)

Matrix

File Name(Linux)

Start

■ 참고 사이트

• BleepingComputer 공식 홈페이지 (https://www.bleepingcomputer.com/news/security/massive-psaux-ransomware-attack-targets-22-000-cyberpanel-instances/)

• SOCRadar 공식 홈페이지 (https://socradar.io/over-22000-cyberpanel-servers-at-risk-from-critical-vulnerabilities-exploitation-by-psaux-ransomware/)

• GitHub (https://gist.github.com/gboddin/d78823245b518edd54bfc2301c5f8882)

• NIST 취약점 데이터베이스 (https://nvd.nist.gov/vuln/detail/CVE-2024-51378)

• BleepingComputer 공식 홈페이지 (https://www.bleepingcomputer.com/news/security/north-korean-govt-hackers-linked-to-play-ransomware-attack/)

• OzarksGo 공식 홈페이지 (https://www.ozarksgo.net/tv-outage-update)

• Unit42 공식 블로그 (https://unit42.paloaltonetworks.com/north-korean-threat-group-play-ransomware/)