정상 소프트웨어 홈페이지로 위장
의심없이 피싱 사이트 접속 및 악성코드 설치 유도

구글 광고를 통해 유포되는 Royal 랜섬웨어

요약

1. 2022년 9월, 외신 블리핑 컴퓨터에 의해 Royal 랜섬웨어 그룹 공개
2. 최초 공격은 2022년 초, Conti 변형 랜섬웨어 사용
3. 2022년 9월, 정상 홈페이지로 위장한 피싱사이트를
   구글광고로 등록, 자체적으로 작성한 Royal 랜섬웨어 노출
4. 피해자에게 공격자 연락처를 제공하기 위한 랜섬노트 본문의 토르 브라우저 URL은
   현재까지도 접속가능, 악성코드 유포는 현재 진행형일 가능성 높음

대응 방안

1. **논리적 망분리를 적용**하여 악성코드 PC 유입을 원천 차단한다
2. **AV+ EDR솔루션을 최신 형상**으로 유지한다.
3. **PC 취약점 주기적으로 점검**하고 **보완**한다.
4. **신뢰할 수 없는 메일의 첨부파일은 실행을 금지**한다.
5. **비 업무 사이트 및 신뢰할 수 없는 웹사이트의 연결을 차단**한다.
6. **OS나 어플리케이션은 최신 형상**을 유지한다.

목차

1. 개요

1.1 배경

1.2 파일 정보

2. 분석

2.1 명령줄 옵션 확인

2.2 섀도 카피본 삭제

2.3 대상 경로 수집

2.4 파일 암호화

2.5 랜섬노트 생성

3. Privacy-i EDR 탐지 정보

4. 대응

1. 개요

1.1 배경

2022년 9월, 외신 블리핑컴퓨터(BleepingComputer)[01] 에 의해 Royal 랜섬웨어 그룹이 수면 위로 드러났다.
최초 공격은 2022년 초부터 시작되었으며 초기에는 기존에 존재하던 랜섬웨어(Blackcat)를 가져다 쓰거나
소스코드가 유출된 랜섬웨어(Conti)를 변형한 것으로 보이는 랜섬웨어를 사용했다.
2022년 9월, 자체적으로 작성한 Royal 랜섬웨어를 다양한 경로를 통해 유포하기 시작했고
마이크로소프트 사[02] 에 의해서도 보고된 바 있다. 그 중에서도 흥미로운 것은 구글 광고를 통한 유포였다.
Royal 랜섬웨어 그룹은 정상 소프트웨어 홈페이지를 위장한 피싱 사이트를 구글 광고에 등록하여 노출시켰다.

[그림 1] 구글애즈(Google Ads) 로고

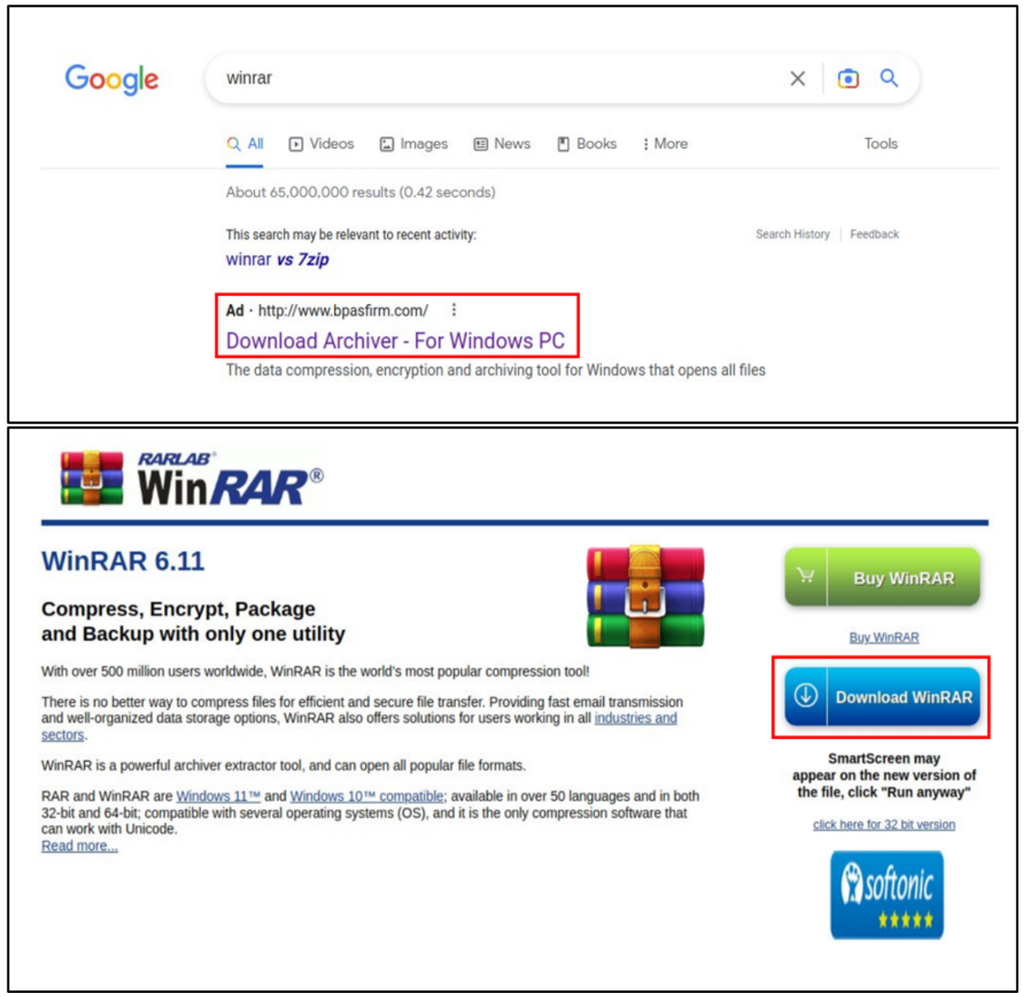
구글 광고 플랫폼인 구글애즈는 광고주가 등록한 키워드를 검색했을 때 검색 결과의 상단에 특정 웹 사이트를 노출시킨다. 이를 악성코드 유포 수단으로 활용했을 때 두 가지 장점이 존재한다.

첫째는 유연한 구매 정책이다. 광고주가 주기적으로 고정 비용을 지불하지 않고 필요에 따라 예산을 정해두면 트래픽 발생 시 예산에서 차감된다. 노출과 별개로 해당 광고 클릭에 따른 비용만이 발생하므로 광고 입찰 경쟁이 덜한 키워드라면 비교적 적은 비용으로 피싱 사이트에 접속하도록 유도할 수 있다.

둘째는 광고 노출 지역 지정이 가능하다. 타이포스쿼팅(정상 도메인을 입력하다가 생기는 오타를 노리는 공격)은 불특정다수를 대상으로 하는 반면, 구글애즈는 특정 지역의 접속자들을 대상으로 설정할 수 있다.

01 https://www.bleepingcomputer.com/news/security/new-royal-ransomware-emerges-in-multi-million-dollar-attacks/

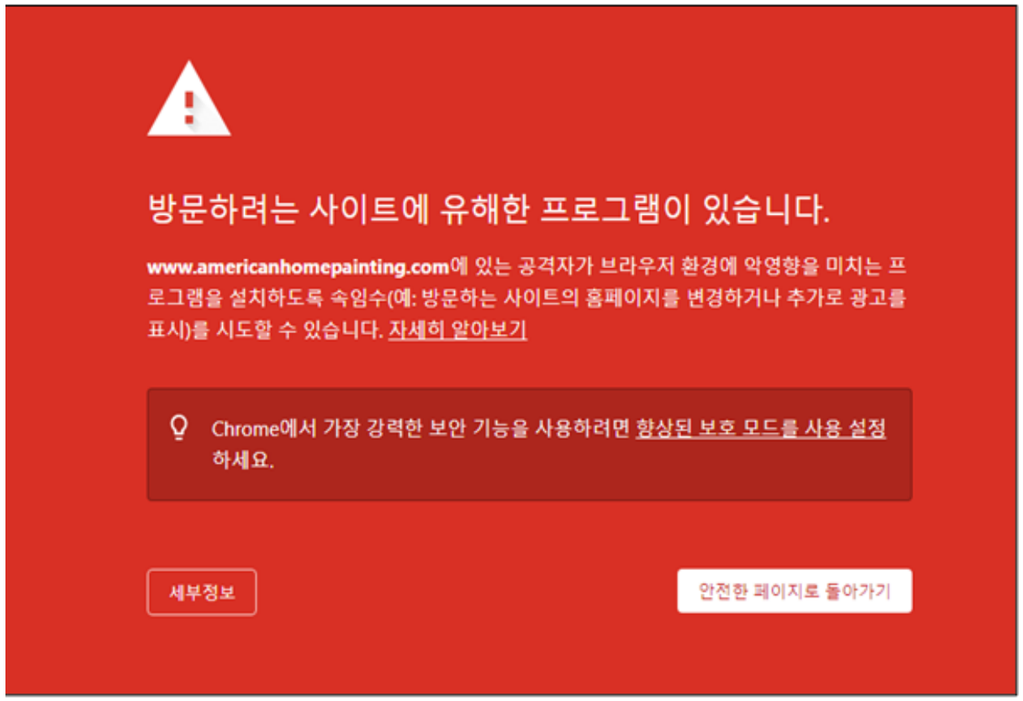
02 https://www.microsoft.com/en-us/security/blog/2022/11/17/dev-0569-finds-new-ways-to-deliver-royal-ransomware-various-payloads/



[그림 2] WinRAR 홈페이지를 사칭한 피싱 사이트 (출처: MalwareHunterTeam 트위터)

위 이미지는 WinRAR(압축 프로그램)를 내려받기 위해 구글 검색을 통해 정상 페이지로 접속한 것처럼 보인다.
하지만 자세히 보면 홈페이지 제목과 웹 사이트의 도메인이 상이한 것을 발견할 수 있다.

도메인에 비해 홈페이지의 제목이 더 눈에 띄므로 피해자는 큰 의심 없이 피싱 사이트로 접속하여
다운로드 버튼을 누른다. 결과적으로 피해자는 WinRAR 뿐만 아니라 악성코드도 함께 설치하게 된다.
다운로드까지 했다면 피해자는 설치 프로그램이 UAC(User Account Control) 프롬프트를 통해
관리자 계정 권한을 요구해도 큰 의심 없이 허용할 확률이 높다. 일반적인 설치 프로그램들이 그렇기 때문이다.



[그림 3] 구글 세이프 브라우징(Google Safe Browsing)의 경고 페이지

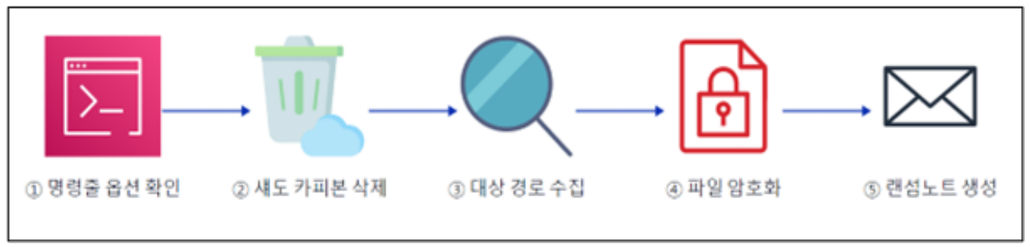
구글은 이러한 악성코드 유포 사이트 접속 시도를 차단하기 위해 악성 URL을 데이터베이스로 관리하고 있다.
만약 사용자가 악성 URL 접속했을 시 경고 페이지를 표시하여 경각심을 불러일으킨다.
이는 웹 브라우저에서 감염을 막을 수 있는 좋은 장치이지만,
신규 도메인은 데이터베이스에 등록되어 있지 않으므로 막기 어렵다는 한계를 가진다.

1.2 파일 정보

Name	windows_encryptor.exe
Type	PE32
Behavior	File encryption
SHA-256	250bcbfa58da3e713b4ca12edef4dc06358e8986cad15928aa30c44fe4596488
Description	Royal Ransomware

[표1] Royal 랜섬웨어 파일 정보

2. 분석



[그림 4] Royal 랜섬웨어 실행 순서도

① 명령줄 옵션 확인
- Royal 랜섬웨어가 지원하는 옵션과 함께 실행하였는지를 확인하기 위해 명령줄에서 파싱한다.

② 섀도 카피본 삭제
- 시스템을 복구하지 못하도록 섀도 카피본을 삭제한다. 그러나 개발자의 실수로 인해 섀도 카피본은 삭제되지 않는다.

③ 대상 경로 수집
- 파일 암호화 대상 경로를 수집한다. 특정 파일 확장자와 폴더는 암호화 대상에서 제외된다.

④ 파일 암호화
- 파일을 사용할 수 없도록 데이터를 암호화한다. 파일 암호화는 AES-256 알고리즘으로 수행하며 대칭키와 IV는 하드코딩 된 RSA 공개키로 암호화된다.

⑤ 랜섬노트 생성
- 피해자에게 연락처를 제공하기 위해 랜섬노트를 생성한다.

2.1 명령줄 옵션 확인

옵션	설명
-id	필수 옵션. 피해자를 식별하기 위한 32글자 길이의 문자열이다. 샌드박스 탐지 우회에 사용되기도 한다.
-ep	선택적 옵션. 파일의 데이터를 대상으로 암호화 정도를 표현하는 백분율. 0부터 100까지만 입력 가능하며 기본값은 50이다.
-path	선택적 옵션. 특정 경로와 그 하위 경로들만 암호화하고 싶을 때 사용한다.

[표 2] Royal 랜섬웨어 옵션 목록

Royal 랜섬웨어가 지원하는 옵션은 총 세 개이다. 반드시 입력해야 하는 필수 옵션 한 개와 필요에 따라 입력하는 선택적 옵션 두 개로 구성되어 있다.

```
if ( lstrlenA(pszOptId) != 32 )
  ExitProcess(0);
```

[그림 5] id 옵션 유효성 검사

id 옵션의 길이는 32여야만 한다. 이를 만족하지 못하면 프로그램은 악성행위를 수행하지 않고 종료된다. 피해자를 식별하기 위한 값임과 동시에 샌드박스 탐지를 우회하는데 사용되기도 한다. id는 감염PC의 특정 정보를 가지고 생성된 SHA-256 해시일 가능성이 높다.

2.2 섀도 카피본 삭제

```
C:\Windows\System32\vssadmin.exe delete shadows /all /quiet
```

[표 3] 섀도 카피본 삭제 명령

섀도 카피본은 특정 시점의 시스템 백업 데이터를 의미하는데, 이를 삭제하면 시스템을 복구하기가 어려워진다. CreateProcessW API로 윈도우즈의 섀도 카피본 관리 프로그램 vssadmin을 실행하여 시스템에 존재하는 섀도 카피본을 모두 삭제한다.

그러나 실제로 섀도 카피본은 삭제되지 않는다. 수집한 Royal 랜섬웨어는 32비트 실행 파일이므로 System32 폴더에 접근 시 WoW64 File System Redirector[03] 에 의해 SysWOW64 폴더로 대신 접근하게 된다. 64비트 윈도우즈 운영체제에서 vssadmin 실행파일은 System32 폴더에만 존재하므로 섀도 카피본 삭제 명령은 실패하게 된다. 이는 개발자의 버그로 추정되며 Wow64DisableWow64FsRedirection API를 호출하여 리다이렉션을 비활성화시키고 System32 폴더에 접근하는 것이 올바른 동작이다.

2.3 대상 경로 수집

```
.exe
.dll
.bat
.lnk
.royal
README.TXT
```

[표 4] 암호화 대상에서 제외되는 파일 확장자 목록

03 https://learn.microsoft.com/en-us/windows/win32/winprog64/file-system-redirector

windows
royal
$recycle.bin
google
perflogs
mozilla
tor browser
boot
$windows.~ws
$windows.~bt
windows.old

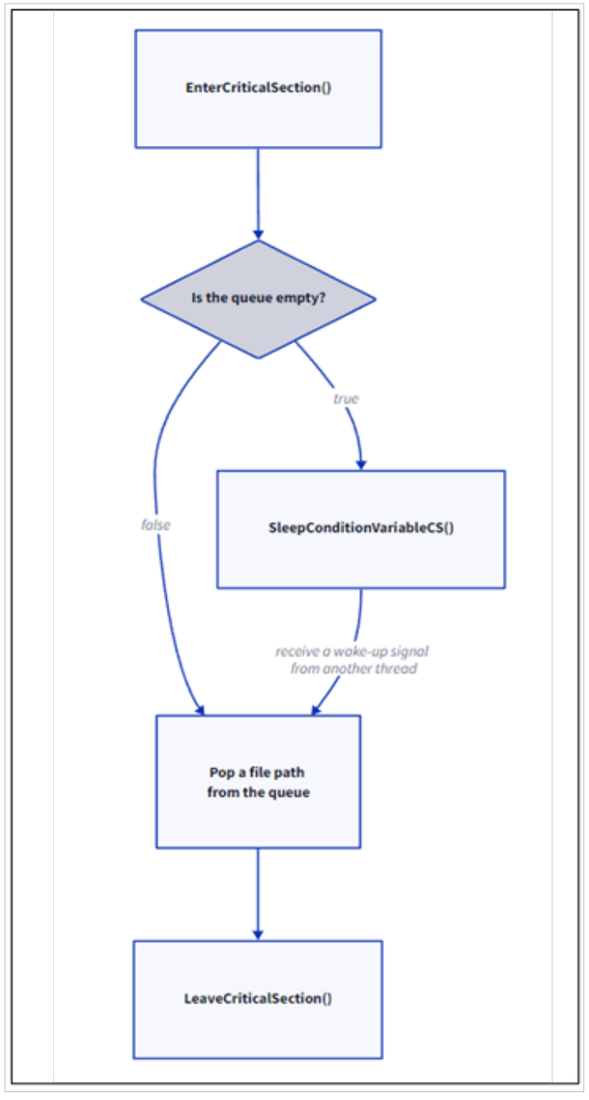
[표 5] 암호화 대상에서 제외되는 폴더 목록

파일 암호화 대상 경로를 수집할 때 제외되는 확장자 및 폴더 목록이다.
예외 처리하는 확장자 목록이 적은 편이므로
사실상 실행파일과 이미 암호화된 파일을 제외한 대부분의 파일이 암호화된다.

```
if ( this->unk_2a0 )
{
  sub_68BFF0(&path, this->pwszOptPath);
  sub_68B880(this, path, v17, v18, v19, (int)v20, p_pFindFileDataW);
}
else
{
  wcscpy(Src, L"A:\\");
  for ( i = GetLogicalDrives(); i; i >>= 1 )
  {
    if ( (i & 1) != 0 )
    {
      path = 0;
      v20 = 0;
      p_pFindFileDataW = 7;
      sub_68C0D0((void **)&path, Src, wcslen(Src));
      sub_68B880(this, path, v17, v18, v19, (int)v20, p_pFindFileDataW);
    }
    ++Src[0];
  }
}
```

[그림 6] 재귀 탐색을 위한 최상위 폴더 선정

수집하는 파일 경로는 path 옵션 여부에 따라 상이하다. 만약, path 옵션이 존재한다면
해당 경로를 최상위 폴더로 기준 삼고 하위의 모든 폴더를 탐색한다.
그게 아니라면 A부터 Z까지의 로컬 드라이브를 최상위 폴더로 기준 삼고 탐색한다.



[그림 7] 파일 암호화 스레드의 파일 경로 획득 순서도

Royal 랜섬웨어는 파일 경로를 수집하는 스레드와 파일을 암호화하는 스레드를 별개로 생성한다.
스레드 동기화는 조건 변수(Condition Variable)로 관리하며 암호화할 파일 경로 수집이 완료되면
WakeAllConditionVariable API를 호출해 다른 스레드에게 작업이 완료됐음을 알린다.
파일 경로 수집은 단일 스레드로 수행하지만,
파일 암호화는 여러 개(CPU 프로세서 개수의 2배) 생성하여 1:N의 관계를 이룬다.

2.4 파일 암호화

암호 관련 코드는 직접 작성하기가 까다롭다.
그래서 대부분의 랜섬웨어는 운영체제에서 지원하는 API를 사용하거나
이미 만들어진 라이브러리를 가져다 사용한다.
Royal 랜섬웨어는 OpenSSL 라이브러리[04]를 가져다 사용하며 버전은 3.0으로 추정된다.

```
hFile = (int)CreateFileW(p_lpFileName, GENERIC_ALL, v16, v17, v18, v19, v20);
if ( hFile == -1 )
{
  LastError = GetLastError();
  if ( LastError == ERROR_SHARING_VIOLATION || LastError == ERROR_LOCK_VIOLATION )
  {
    sub_44BFF0(&v15, &lpFileName);
    if ( !ActiveRestartManager(v15, v16, (int)v17, v18, v19, (unsigned int)v20) )
      goto FAIL;
  }
  else if ( LastError == ERROR_ACCESS_DENIED )
  {
    goto FAIL;
  }
```

```
SHUTDOWN_PROCESS:
        v14 = RmShutdown(pSessionHandle, RmForceShutdown, 0);
        free(Block);
        RmEndSession(pSessionHandle);
        v15 = v14 == 0;
        goto LABEL_27;
      }
      v13 = Block;
      while ( v13->Process.dwProcessId != dwCurPid && v13->Process.dwProcessId != dwExplorerPid )
      {
        ++v12;
        ++v13;
        if ( v12 >= pnProcInfo )
          goto SHUTDOWN_PROCESS;
      }
```

[그림 8] 파일 접근 불가 시 Restart Manager로 우회

암호화 대상 파일에 접근하기 위해 파일 핸들을 얻는다.
이 때, 다른 프로세스가 해당 파일을 사용 중(ERROR_SHARING_VIOLATION)이거나
또는 파일의 일부를 잠금하여 단독 액세스 권한을 부여받은 경우(ERROR_LOCK_VIOLATION),
Restart Manager[05] 를 통해 해당 프로세스를 종료시킨다.

이 때, 랜섬웨어 자기자신과 파일 탐색기(explorer.exe)는 강제 종료 대상에서 제외시킨다.
본래 Restart Manager는 소프트웨어 업데이트 시 이미 실행 중인 프로세스에 의해
업데이트 관련 파일에 접근하지 못할 때 사용하도록 만들어졌지만,
이처럼 랜섬웨어가 더 많은 파일을 암호화하기 위해 악용하기도 한다.

04 https://github.com/openssl/openssl

05 https://learn.microsoft.com/en-us/windows/win32/rstmgr/about-restart-manager

마지막으로, 권한이 부족해 접근이 불가한 파일은 암호화를 생략한다.

```
BIO_write(
  v2,
  "-----BEGIN RSA PUBLIC KEY-----\n"
  "MIICCAKCAgEA4obXj0xaFm0kiaM+FTz8tdYy7AxyZlohP3bjvyXSl4nfQe4nz5c4\n"
  "3KQqdaRrNsYrwKdiUg7T4MkTMuDFiA7/O9zGBQUGohT+wQ8v7xVhMHO5baSe0Zt3\n"
  "3MOj+NU71N705rYMEe2RsSXDNKc63h1/VPD04FI54kaMlzQoD7FaunViRDfVwUNP\n"
  "VW5ToEwwI/CxkKevPyYTgaI1hkx4hCj+JLrxBt/8dBoySDlsYJijIRhvvBBPQN/p\n"
  "NRM1X05gYffhL+pLRtVOvKsJ4qurQUWaK9zxFSg+fIdDRCYR1+updXAGHpQccahe\n"
  "OaIfob+y8+mLJtnpW7Zxu8wVbUsrzoKCQOn3iNohIuunHz8mDiCLdkvtAC+vh9a6\n"
  "rdxgQ+U9n3xJy1bwD7MSgoeIgWR9+bxhTjoP1WtmqyH3aESeFvVzqSao1fWhmVBx\n"
  "XuIHROKwXU+QYwNh6OlRuvPZ5KTjqhTX8wuJXuY4fwCOGn706jm5X4BKpXIzcWiS\n"
  "ysYlidPzoa+jpPCAFPwKBbR1knKrZeObkFPvZZ1SFDTz9ZSK9/WSg4w8a7RfbF4g\n"
  "c1J7GnXaqvKrN6eHDWG5F6GekVGzEXBDU8CEa15EA0jcKacqMy90huA2YJciQ4Gl\n"
  "WvywwZ/NRpa0QOJr4Z2DjH88jXXeE583Ls1ugl04f1NuvfF/x5mITR8CAQM=\n"
  "-----END RSA PUBLIC KEY-----\n"
  "\r\n",
  pkl);
```

[그림 9] 하드코딩 된 RSA 공개키

```
RAND_bytes_ex(key, 32);                    // generate a random AES-256 key
RAND_bytes_ex(&iv, 16);                    // generate a random AES-256 IV
*(_OWORD *)v60 = key[0];
v61 = key[1];
v62 = iv;
RSA_public_encrypt(48, v60, v60, rsa, 4);
```

[그림 10] 무작위 AES 대칭키와 IV 생성 후 RSA 암호화

공격자의 RSA 공개키는 바이너리에 하드코딩되어 있다.
공개키는 파일의 암호화를 위해 무작위로 생성된 AES-256의 대칭키와
IV(Initialize Vector)를 암호화하는 데 쓰인다. 대칭키와 IV는 매 파일마다 무작위로 생성된다.

```
AES_cbc_encrypt(out, out, inl, aes, &iv, 1);
v37.HighPart = -(inl != 0);
v37.LowPart = -inl;
SetFilePointerEx(hFile, v37, 0, SEEK_CUR);
v27 = 0;
v28 = 0;
while ( 1 )
{
  NumberOfBytesRead[0] = 0;
  if ( !WriteFile(hFile, (char *)out + v27, inl - v28, NumberOfBytesRead, 0)
    goto LABEL_33;
```

[그림 11] AES 암호화되는 파일 데이터

파일 암호화는 AES-256 CBC(Cipher Block Chaining) 모드로 수행한다.
CBC는 블록 암호에서 가장 많이 사용되며 블록 모드 중 보안성이 가장 높다.
최초 블록을 암호화하는 데 쓰인 IV를 알지 못하면 다른 블록들도 복호화가 어렵기 때문이다.

```
00001E80  CC 2D 8A 26 53 79 D1 DF 9C AD 83 B6 6C 62 35 D0
00001E90  76 23 87 FE C2 C0 AD 95 B8 EB 1C 1A 9C BF F5 7D
00001EA0  C9 A1 96 13 12 AC 12 E9 F0 E8 50 C6 73 CD 02 1A
00001EB0  41 78 7D A0 80 FE CA 9B D6 2F 32 7A 3C A5 9C B9
00001EC0  13 31 78 DF 06 58 E2 22 AA D3 57 17 82 B8 15 D6
00001ED0  DC D2 AB D4 3D 53 E6 61 59 40 26 AC A6 5F 63 EF
00001EE0  EC 67 47 99 1A ED 3C 55 A4 97 AB E0 79 ED 6C 9A
00001EF0  4A F7 BA 6D E6 5F CA 0A 89 FB 33 65 EC 18 0B 5D
00001F00  3A D9 09 EB B7 D1 B4 A3 95 5D 61 07 A3 CB 7F 5A
00001F10  33 9A E3 D2 F9 BA 70 48 F1 DD 37 15 5D 9F B7 38
00001F20  C0 96 17 B4 F5 2D 0C 97 E2 C4 2D E8 99 B3 63 E9
00001F30  A1 2E 5B B0 C9 3A 74 3B 4C 83 AD D0 32 C0 DC 0A
00001F40  02 41 93 6C C9 10 4D EC B4 85 A5 1D 50 B3 DF CD
00001F50  42 1D 00 00 00 00 00 00 64 00 00 00 00 00 00 00
```

[그림12] 복호화를 위한 메타데이터 정보

오프셋	크기	설명
-512	512	RSA 공개키로 암호화된 무작위 AES 대칭키와 IV
-16	8	원본 파일 크기
-8	8	암호화 정도 백분율

[표 6] 복호화를 위한 메타데이터 정보

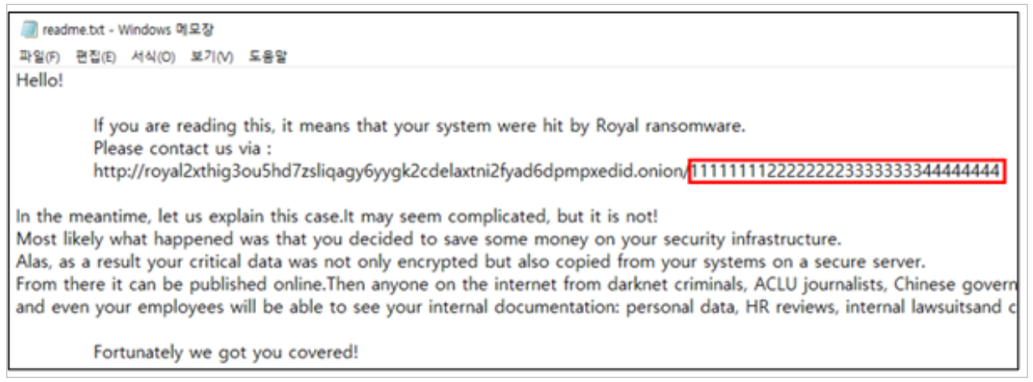
암호화된 파일의 마지막엔 파일 복호화를 위한 공격자의 정보만이 추가된다.
추가되는 정보에는 암호화된 무작위 AES 대칭키와 IV, 원본 파일 크기,
그리고 파일 데이터의 암호화 정도를 나타내는 백분율 등이 있다.


```
if ( bEncrypted )
{
  sub_68C230(v16, v17, L".royal");
  v13 = 8;
  v12 = (LPCWSTR)std::wstring::c_str(v16);
  v9 = (const WCHAR *)std::wstring::c_str(v17);
  MoveFileExW(v9, v12, v13);
  std::wstring::~wstring(v16);
}
```

[그림13] 암호화된 파일에 확장자 추가

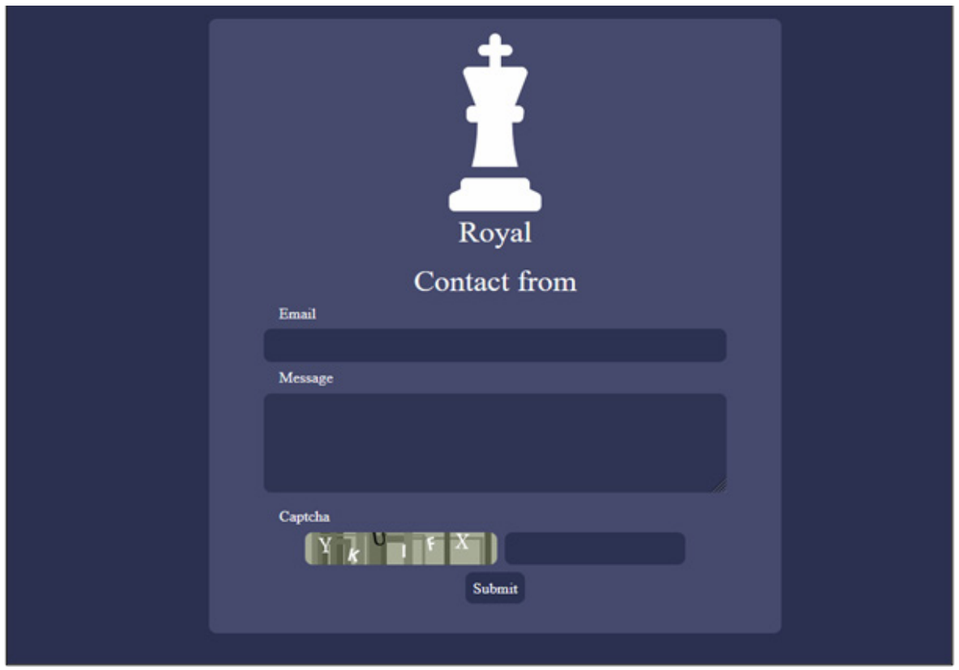
파일 암호화가 성공적으로 됐으면 파일명의 마지막에 전용 확장자(.royal)를 붙여 변경한다.

2.5 랜섬노트생성



[그림 14] 랜섬노트 본문

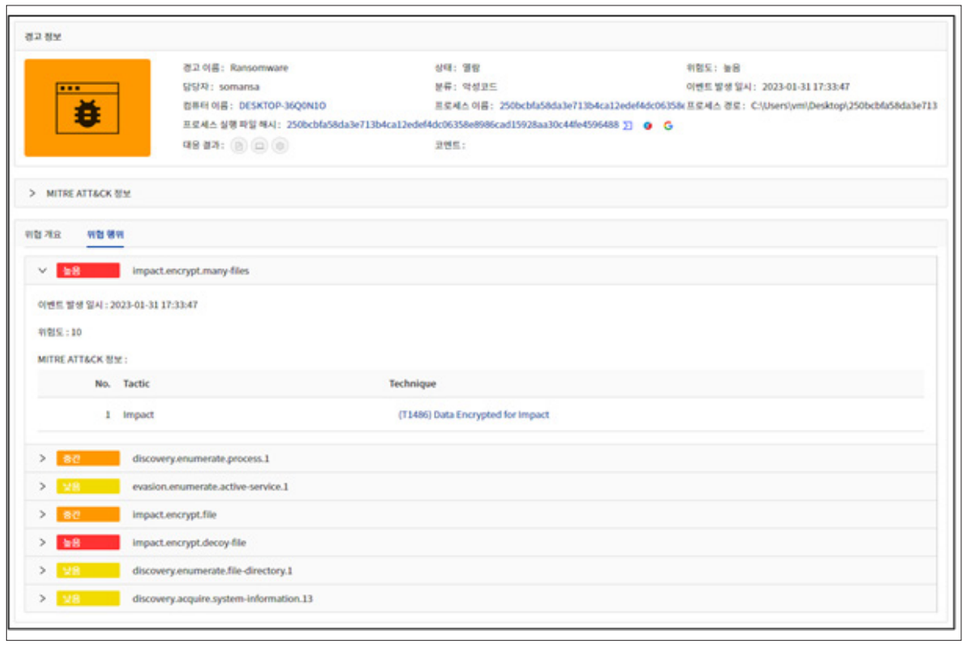
랜섬노트에는 피해자를 위한 공격자의 연락처(토르 브라우저 링크)가 있다.
URL의 마지막엔 최초 랜섬웨어 실행 시 입력했던 id 옵션값이 포함되어 있다.
랜섬노트 본문은 바이너리에 하드코딩되어 있으며,파일 경로를 수집하는 스레드가 랜섬노트 파일을 생성한다.
파일명은 항상 README.TXT이다.



[그림 15] 토르 URL 접속 페이지 화면

랜섬노트 본문의 토르 브라우저 URL은 현재까지도 접속이 가능하다.
따라서 악성코드 유포가 현재진행형일 확률이 높다

3. Privacy-i EDR 탐지 정보



[그림 16] Privacy-i EDR 탐지 정보

Privacy-i EDR은 Royal 랜섬웨어의 파일 암호화 행위를 탐지하여 랜섬웨어로 분류하고 있다.
볼륨 섀도 카피본 삭제는 실질적인 행위가 이루어지지 않았으므로 탐지 또한 되지 않았다.

4. 대응

1. 논리적 망분리를 적용하여 악성코드 PC 유입을 원천 차단한다
2. AV+ EDR솔루션을 최신 형상으로 유지한다.
3. PC 취약점 주기적으로 점검하고 보완한다.
4. 신뢰할 수 없는 메일의 첨부파일은 실행을 금지한다.
5. 비 업무 사이트 및 신뢰할 수 없는 웹사이트의 연결을 차단한다.
6. OS나 어플리케이션은 최신 형상을 유지한다.

본 자료의 전체 혹은 일부를 소만사의 허락을 받지 않고, 무단게재, 복사, 배포는 엄격히 금합니다.
만일 이를 어길 시에는 민형사상의 손해배상에 처해질 수 있습니다.
본 자료는 악성코드 분석을 위한 참조 자료로 활용 되어야 하며,
악성코드 제작 등의 용도로 악용되어서는 안됩니다.
㈜ 소만사는 이러한 오남용에 대한 책임을 지지 않습니다.
Copyright(c) 2023 ㈜ 소만사 All rights reserved.

궁금하신 점이나 문의사항은 malware@somansa.com 으로 문의주십시오