랜섬웨어 암호기능 분석 보고서

- 5ss5c -

2020. 12

차세대암호인증팀

KISA 한국인터넷진흥원
KOREA INTERNET & SECURITY AGENCY

Contents

1. 개요

5ss5c 랜섬웨어는 2019년 11월경부터 탐지되었고, 2020년 1분기 국내에 유포되어 피해를 입힌 랜섬웨어이다. 유포방식과 코드 제작 방식이 2018년부터 유포된 Satan 랜섬웨어와 비슷하여 제작자가 동일하거나, Satan 랜섬웨어 소스코드를 기반으로 작성한 것으로 추측된다. 5ss5c는 다운로더를 통해 악성 프로세스를 동작시키고, ExternalBlue 익스플로잇을 사용하여 유포하고 있다. 이는 과거 DBGer, Lucky, Iron 악성코드 등을 유포한 집단과 관련이 있는 것으로 추정된다.

본 보고서에 사용된 5ss5c 샘플의 해시값은 다음 [표 1]과 같다.

랜섬웨어 샘플 파일 해시값	
MD5	853358339279B590FB1C40C3DC0CDB72
SHA1	84825801EAC21A8D6EB060DDD8A0CD902DCEAD25
SHA256	CA154FA6FF0D1EBC786B4EA89CEFAE022E05497D095C2391331F24113AA31E3C

[표 1] 5ss5c 랜섬웨어 샘플 해시값

2. 랜섬웨어 분석

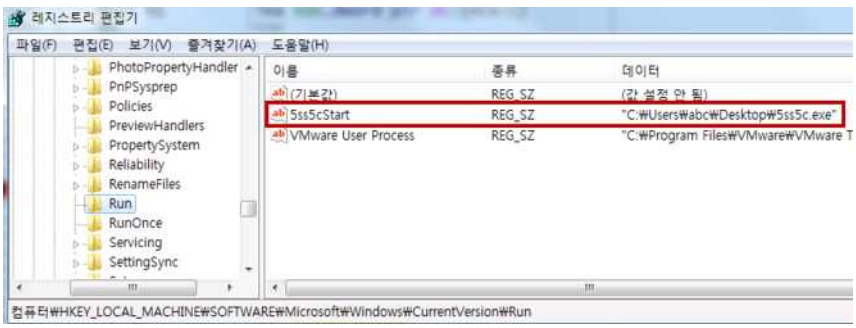
2.1 실행 과정

5ss5c 랜섬웨어의 전체 동작 과정은 [그림 1]과 같다.

[그림 1] 5ss5c 동작 과정

(1) 레지스트리 등록

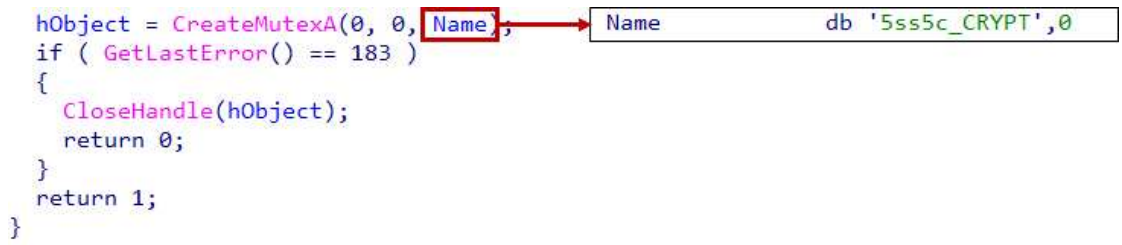
랜섬웨어가 동작되면 랜섬웨어 실행 경로를 [그림 2]와 같이 레지스트리 “HKLM\Software\MicroSoft\Windows\CurrentVersion\Run” 경로에 등록하여 PC를 부팅할 때마다 자동으로 랜섬웨어가 실행되도록 설정한다.



[그림 2] 레지스트리에 등록된 5ss5c 프로그램 정보

(2) 뮤텍스 생성

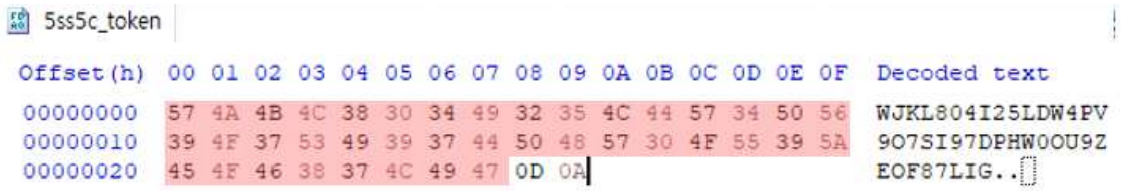
랜섬웨어 실행 시 쓰레드가 중복되어 랜섬웨어가 동작하지 않는 상황을 방지하기 위하여 [그림 3]과 같이 “5ss5c_CRYPT” 라는 이름의 뮤텍스를 생성한다.



[그림 3] 뮤텍스 생성

(3) 토큰 생성

네트워크 통신 중 감염 PC 사용자 식별을 위해 40 bytes 크기의 토큰을 생성한다. 0-9와 A-Z까지 중에서 **40개**의 문자열을 무작위로 생성한다. [그림 4]는 샘플에서 생성된 토큰을 보여준다. 생성 후 감염 PC의 "C:\ProgramData\5ss5c_token" 경로에 저장하여 사용한다. 또한 이 때 생성한 토큰을 암호화된 파일명과 랜섬노트에 기재한다.



[그림 4] 생성된 토큰

(4) 암호키 생성

랜섬웨어 파일 내부에 하드코딩된 메르센 트위스터(Mersenne Twister)[1] 알고리즘을 통해 200 bytes 크기의 랜덤 문자열을 생성한다. 이후 [그림 5]와 같이 랜섬웨어 파일 내부 바이너리 0x5E06DC 위치에 하드코딩된 16 bytes 문자열과 메르센 트위스터에서 생성한 200 bytes 문자열 중 상위 16 bytes를 연접한 32bytes를 암호키로 사용한다.

[그림 5] 하드코딩된 문자열

(5) 특정 프로세스 및 서비스 종료

파일 암호화를 진행하기 전 [표 2]와 같이 데이터베이스와 관련된 특정 프로세스 및 서비스를 종료한다.

1) 난수의 반복 주기가 메르센 소수($2^n - 1$)인 유사난수 생성기

프로세스 종료 목록			
Sql	oracle	sqlservr.exe	mysqld.exe
sqlagent.exe	fdhost.exe	fdlauncher.exe	reportingservicesservice.exe
tnslsnr.exe	oracle.exe	emagent.exe	perl.exe
mysqld-nt.exe	nmesrvc.exe	omtsreco.exe	sqlwriter.exe
서비스 종료 목록			
MySQL	MySQLa	SQLWriter	SQLSERVERAGENT
UxSms	MSSQLFDLauncher		

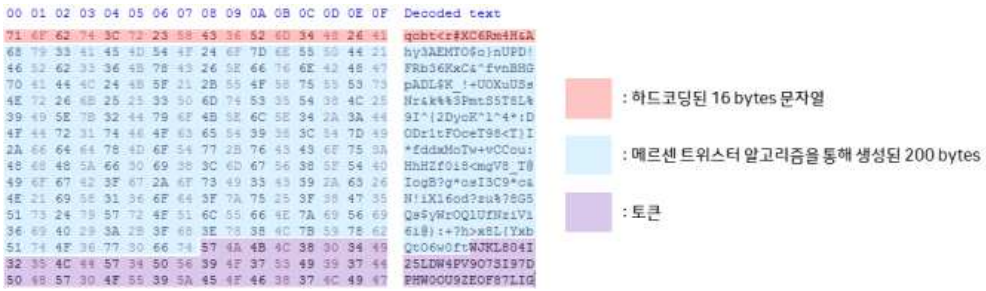
[표 2] 프로세스 종료 목록

(6) 랜섬노트 생성

랜섬노트는 감염된 PC의 “C:\” 경로에 중국어로 작성된다. 랜섬노트에는 일부 파일이 암호화되었다는 문구와 함께 1비트코인을 요구하는 내용이 포함되어 있다. 저장되는 값의 생성과정은 [그림 6]과 같다.

[그림 6] 5ss5c 랜섬노트 생성과정

랜섬노트에 저장되는 값 생성과정은 다음과 같다. 암호키 생성 시 사용되는 하드코딩된 16 bytes의 문자열, (4)에서 메르센 트위스터 알고리즘으로 생성한 랜덤 200 bytes와 (3)에서 생성한 토큰을 [그림 7]과 같이 연접한 후에 랜섬웨어 파일 내에 하드코딩된 RSA-4096 공개키를 사용해 암호화한다. 암호화된 바이너리 값을 헥사 스트링([그림 6]의 A)으로 변환한다.

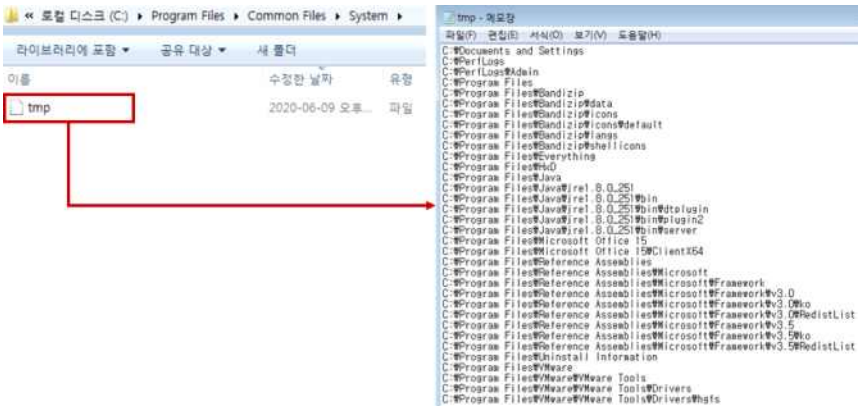


[그림 7] 랜섬노트에 저장하기 위한 데이터

이후에 ((A[i] XOR 0x19) – 0x7A)와 같은 수식으로 바이트 별로 연산 과정을 거친다. 이때 A[i]는 A를 순서대로 1byte씩 가져온 값을 의미한다. 최종적으로 base64로 인코딩된 값을 랜섬노트에 저장한다.

(7) 파일 암호화

5ss5c 랜섬웨어는 [그림 8]과 같이 “C:\ProgramFiles\CommonFiles\System” 경로의 tmp 파일에 감염시킨 컴퓨터의 디스크에서 암호화 할 디렉터리를 선정하여 저장한다.



[그림 8] tmp 파일에 저장된 암호화 대상 디렉터리

암호화 제외 대상 파일 확장자 및 디렉터리는 [표 3]과 같다. 윈도우 및 시스템 관련 디렉터리뿐만 아니라 중국의 인터넷 보안 회사인 Qihoo 360과 관련된 360safe와 360downloads과 같은 디렉터리는 암호화하지 않는다.

암호화 제외 대상 확장자 목록	암호화 제외 디렉터리
bin, bmp, cab, chm, dat, dll, exe, iso, lib, log, msi, ocx, pbk, pol, sdi, sys, tmp, wim	360rec, 360sec, 360sand, 360safe, 360downloads, favorites, common files, default user, all users, libs, internet explorer, msbuild, public, windows mail, windows media player, windows defender, windows nt, windows photo viewer, windows sidebar, temp

[표 3] 암호화 제외 대상 확장자 및 디렉터리 목록

(4)에서 생성한 암호키를 사용하여 감염 대상 파일을 AES-256-ECB 알고리즘으로 암호화를 진행한다. 암호화 후 파일명은 '[5ss5c@mail.ru]원본 파일명.원본 파일 확장자.토큰.5ss5c'으로 변경된다. 이때 토큰은 (3)에서 생성된 문자열이다.

2.2 암호 키 생성 과정

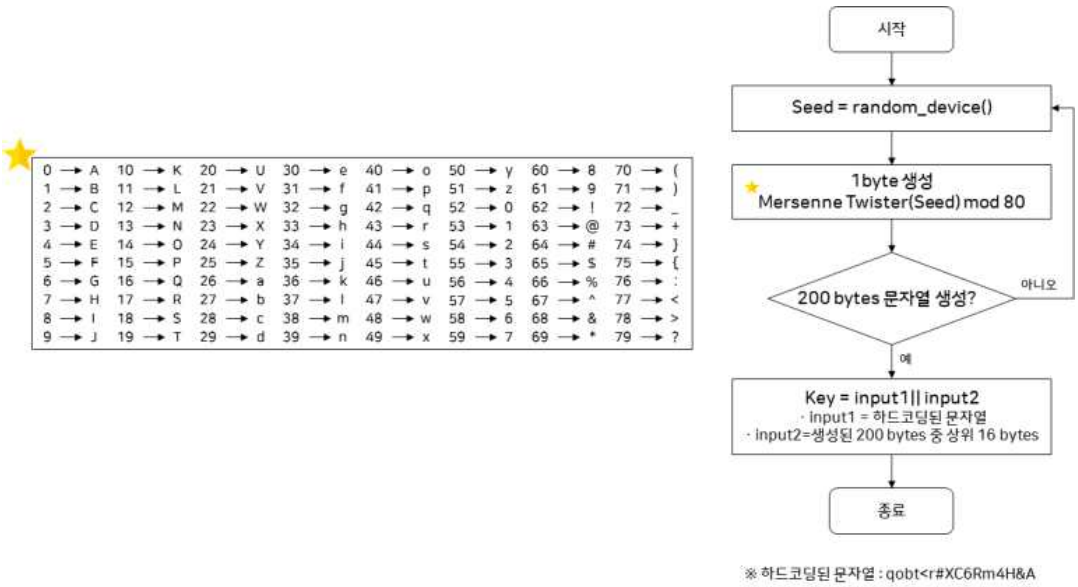
파일 암호키 생성과정은 [그림 10]와 같다. 먼저, C++의 기본 라이브러리인 Numerics library의 Pseudo -random number generation 클래스 내 random_device 함수를 사용하여 32bytes의 seed를 생성한다. 이때 사용하는 value는 "rand_s"이며, [그림 9]와 같다.

```
uint __cdecl _Random_device(void)

{
  int iVar1;
  uint local_8;

  iVar1 = _rand_s(&local_8);
  if (iVar1 != 0) {
    _Xout_of_range(s_invalid_random_device_value_005b4404);
  }
  return local_8;
}
```

[그림 9] random_device 함수 value

seed를 이용하여 메르센 트위스터 알고리즘을 통해 난수를 생성한다. 생성한 난수는 모듈러 80 연산을 통해 0에서 79 사이의 정수로 선택한 후 ASCII 값으로 대응하여 총 200 bytes의 문자열을 생성한다. 마지막으로 고정된

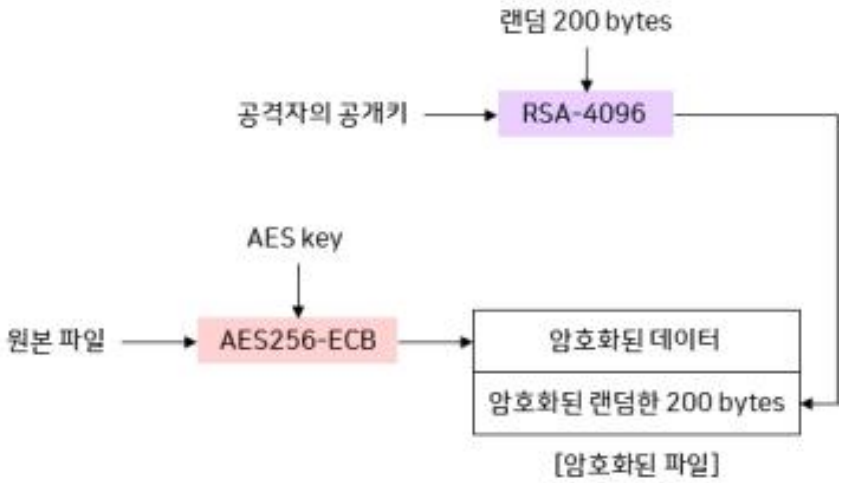
문자열인 16bytes의 “qobt<r#XC6Rm4H&A” 와 메르센트위스터에서 생성한 200 bytes 중 상위 16 bytes를 연접한 32bytes를 암호키로 사용한다.



[그림 10] 암호키 생성과정

2.3 암호화 과정

5ss5c 랜섬웨어의 파일 암호화 과정은 [그림 11]와 같다. 먼저, 피해자의 파일을 AES-256-ECB 암호키를 생성해 암호화한다. 그리고 피해자의 PC에서 RSA-4096 공개키쌍을 생성한 뒤, RSA-4096 의 개인키로 각 파일을 암호화 하는데 쓰인 AES 암호키를 암호화하여 암호화한 데이터에 하단에 덧붙인다. 마지막으로 피해자의 PC에서 생성한 RSA-4096 개인키를 공격자가 랜섬웨어 내부에 하드코딩한 RSA-4096 공개키로 암호화한다.



[그림 11] 암호화 과정

암호화는 패딩 알고리즘 없이 AES-256-ECB를 사용한다. 따라서 마지막 블록 데이터가 16 bytes 미만인 경우 마지막 블록은 암호화를 하지 않고 원본 데이터를 그대로 저장한다.

암호화된 파일의 구조는 [그림 12]과 같다. 암호화된 데이터 뒤에 위의 과정에서 생성한 200 bytes 문자열이 하드 코딩된 공개키를 사용해 RSA-4096으로 암호화되어 저장된다.

[그림 12] 암호화된 파일 구조

3. 복구 시나리오

(1) 제작자의 RSA-4096 개인키를 획득하는 경우

제작자의 RSA-4096 개인키를 획득하면 랜섬노트에 제작자의 공개키로 암호화된 RSA-4096 개인키 데이터(피해자의 PC에서 생성한)를 복호화할 수 있다. 파일 끝에 저장된 랜덤한 200 bytes를 복호화할 수 있으므로 고정 문자열인 “qobt<r#XC6Rm4H&A”와 연접하여 AES 파일 암호키를 획득할 수 있다.

(2) 파일 암호키를 재현할 수 있는 경우

AES 파일 암호키 재현을 위해서는 랜섬웨어 실행 당시의 하드웨어 리소스 값에 대한 정보를 얻어야 한다. 5ss5c는 메르센 트위스터 알고리즘을 200회 반복하므로 200개의 하드웨어 리소스 값을 획득해야 한다. 해당 값을 seed로 하여 메르센 트위스터 알고리즘을 통해 키 생성에 필요한 값을 획득할 수 있다.

또는, AES 파일 암호화 키의 32 bytes 중 고정값인 16 bytes를 제외하고 랜덤한 16 bytes를 전수조사하면 키를 획득할 수 있다. 이 중 두 번째 방법의 전수조사량이 2^{101}이므로 첫 번째 방법의 $(2^{32})^{200}$보다 더 작지만 이 또한 재현 불가능한 수치이다.

4. 결론

5ss5c 랜섬웨어의 분석 결과를 정리한 내용은 [표 4]와 같다. 파일 암호화에 사용된 암호 알고리즘은 AES-256-ECB이며, 파일 암호키는 해당 알고리즘에서 생성된 16bytes와 하드코딩된 16bytes를 사용하여 생성한다. 파일 암호화에 사용된 키들은 사용이 끝나면 메모리 내에서 제로화 및 할당 해제하므로 키 정보를 얻어내는 것이 불가능하다.

파일 암호키 획득 및 재현은 불가능하지만 5ss5c는 볼륨 섀도우 복사본을 삭제하지 않기 때문에 감염 시점 이전의 파일을 저장하고 있는 볼륨 섀도우 복사본이 존재하는 경우 원본 파일을 획득할 수 있다. Shadow Explorer 프로그램을 통해 복원 지점의 파일을 획득할 수 있다.

제작자의 RSA-4096 개인키를 획득하거나 파일을 암호화하는 AES 암호키를 재현할 수 있는 경우에 랜섬웨어 복구가 가능하다.

분석 항목	상세 요소	분석 내용
암호학적 요소	암호 알고리즘	AES256-ECB (파일 암호화) RSA-4096 (파일 암호키 암호화)
	고정 키 유무	X
	모든 파일 동일한 암호키	O
	IV, salt 값	X
	키 생성 seed	random_device
	키 생성 방법	Mersenne Twister Algorithm
메모리 요소	할당 해제 수행 여부	O
	제로화 수행 여부	O
시스템	볼륨 섀도 복사본 삭제	X

[표 4] 5ss5c 랜섬웨어 분석 항목 및 결과

랜섬웨어 암호기능 분석 보고서 작성 공헌자

구 분	소 속	직 위	성 명
책임자	KISA	팀장	박창열
작성자	KISA	책임	김기문
		선임	김대운

랜섬웨어 암호기능 분석 보고서 - 5ss5c

2020년 12월 발행

발행처 KISA 한국인터넷진흥원
KOREA INTERNET & SECURITY AGENCY