RansomEXX Trojan attacks Linux systems

By Fedor Sinitsyn

Published: 2020-11-06 · Archived: 2026-04-05 21:37:11 UTC

We recently discovered a new file-encrypting Trojan built as an ELF executable and intended to encrypt data on machines controlled by Linux-based operating systems.

After the initial analysis we noticed similarities in the code of the Trojan, the text of the ransom notes and the general approach to extortion, which suggested that we had in fact encountered a Linux build of the previously known ransomware family RansomEXX. This malware is notorious for attacking large organizations and was most active earlier this year.

RansomEXX is a highly targeted Trojan. Each sample of the malware contains a hardcoded name of the victim organization. Moreover, both the encrypted file extension and the email address for contacting the extortionists make use of the victim's name.

Several companies have fallen victim to this malware in recent months, including the Texas Department of Transportation (TxDOT) and Konica Minolta.

Technical description

The sample we came across – aa1ddf0c8312349be614ff43e80a262f – is a 64-bit ELF executable. The Trojan implements its cryptographic scheme using functions from the open-source library mbedtls.

When launched, the Trojan generates a 256-bit key and uses it to encrypt all the files belonging to the victim that it can reach using the AES block cipher in ECB mode. The AES key is encrypted by a public RSA-4096 key embedded in the Trojan's body and appended to each encrypted file.

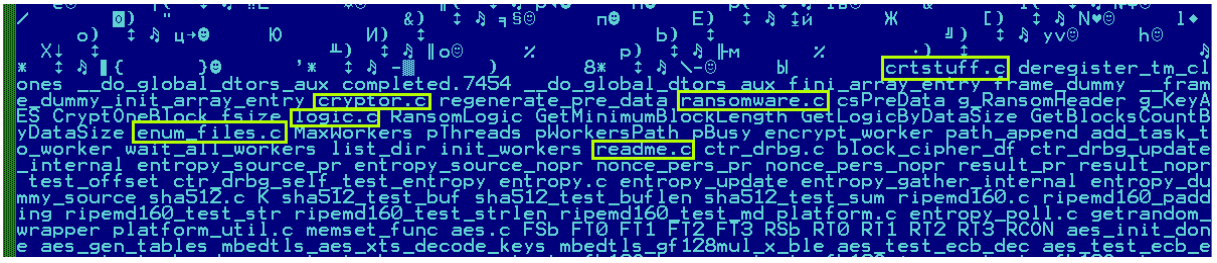
Additionally, the malware launches a thread that regenerates and re-encrypts the AES key every 0.18 seconds. However, based on an analysis of the implementation, the keys actually only differ every second.

Apart from encrypting the files and leaving ransom notes, the sample has none of the additional functionality that other threat actors tend to use in their Trojans: no C&C communication, no termination of running processes, no anti-analysis tricks, etc.

```
if ( a1 )
{
  v10 = strlen(s) + 277;
  v2 = alloca(16 * ((v10 + 23LL) / 0x10uLL));
  dest = (16 * ((&s + 7) >> 4));
  if ( dest )
  {
    strcpy(dest, s);
    strcat(dest, ".[redacted]");
    v3 = rand();
    sprintf(src, "-%08x", v3);
    strcat(dest, src);
    if ( fsize(dest) == -1 )
    {
      stream = fopen64(s, "r+");
      if ( stream )
      {
        v9 = fsize(s);
        if ( v9 )
        {
          if ( v9 > 15 )
          {
            mbedtls_aes_init(aes_ctx);
            pthread_mutex_lock(&csPreData);
            qmemcpy(ptr, &g_RansomHeader, 0x200uLL);
            mbedtls_aes_setkey_enc(aes_ctx, &g_KeyAES, 256LL);
            pthread_mutex_unlock(&csPreData);
            if ( !fseek(stream, 0LL, SEEK_END)
              && fwrite(ptr, 1uLL, 0x200uLL, stream)
              && !fseek(stream, -512 - v9, 1)
              && ProcessFileHandleWithLogic(stream, aes_ctx, a2, v9, CryptOneBlock) )
            {
              v13 = 1;
            }
```

Fragment of the file encryption procedure pseudocode; variable and function names are saved in the debug information and must match the original source code

Curiously, the ELF binary contains some debug information, including names of functions, global variables and source code files used by the malware developers.



Original names of source files embedded in the trojan's body

```
[06ac][06ac] << clone (,,,,) [00000000000006b1] {0000}
[06ac][06b1] << clone (,,,,) [0000000000000000] {0000}
[06ac][06b1] >> set_robust_list (00007fa5e1ef49e0,0000000000000018) => 00007fa5f42cee44 {0000}
[06ac][06b1] << set_robust_list (,) [00000000] {0000}
[06ac][06b1] >> sched_setaffinity (00000000,00000080,00007fa5e1ef3e70 -> 0000000000000008) => 00007fa5f3fdc8ff {0000}
[06ac][06ac] >> openat (ffffffffffffff9c -> "",000055a3553bc0a0 -> "/root",00090800,0000) => 00007fa5f3fe8820 {0000}
[06ac][06ac] << openat (,,,) [0000000000000003 -> {Directory:"/root"}] {0000}
[06ac][06ac] >> stat (000055a3553bc0c0 -> "/root/!NEWS_FOR_ .!.txt",00007ffc35408c60) => 00007fa5f3fe82c5 {0000}
[06ac][06ac] << stat (,00007ffc35408c60 -> (struct stat*){st_dev=0000000000000000,st_ino=0000000000000000,st_nlink=0000000000000000,st_mode=00000000,st_uid=0
[06ac][06ac] >> open (000055a3553bc0c0 -> "/root/!NEWS_FOR_ .!.txt",00000241,01b6) => 00007fa5f3fe877d {0000}
[06ac][06ac] << open (,,) [0000000000000004 -> {File:"/root/!NEWS_FOR_ .!.txt"}] {0000}
[06ac][06ac] >> fstat (0000000000000004 -> {File:"/root/!NEWS_FOR_ .!.txt"},00007ffc35408ad0) => 00007fa5f3fe8314 {0000}
[06ac][06ac] << fstat (,00007ffc35408ad0 -> (struct stat*){st_dev=000000000000fd00,st_ino=0000000006127b79,st_nlink=0000000000000001,st_mode=000081a4,st_uid=
[06ac][06ac] >> mmap (0000000000000000,0000000000001000,00000003,00000022,ffffffffffffffff -> "",0000000000000000) => 00007fa5f3ff1d9a {0000}
[06ac][06ac] << mmap (,,,,,) [00007fa5f4701000] {0000}
[06ac][06ac] >> write (0000000000000004 -> {File:"/root/!NEWS_FOR_ .!.txt"},00007fa5f4701000 -> "Greetings .!!!\r\n\r\nStudy this message REGARDFULLY
[06ac][06ac] << write (,,) [00000000000001ec] {0000}
[06ac][06ac] >> close (0000000000000004 -> {File:"/root/!NEWS_FOR_ .!.txt"}) => 00007fa5f3fe9050 {0000}
[06ac][06ac] << close () [0000000000000000] {0000}
```

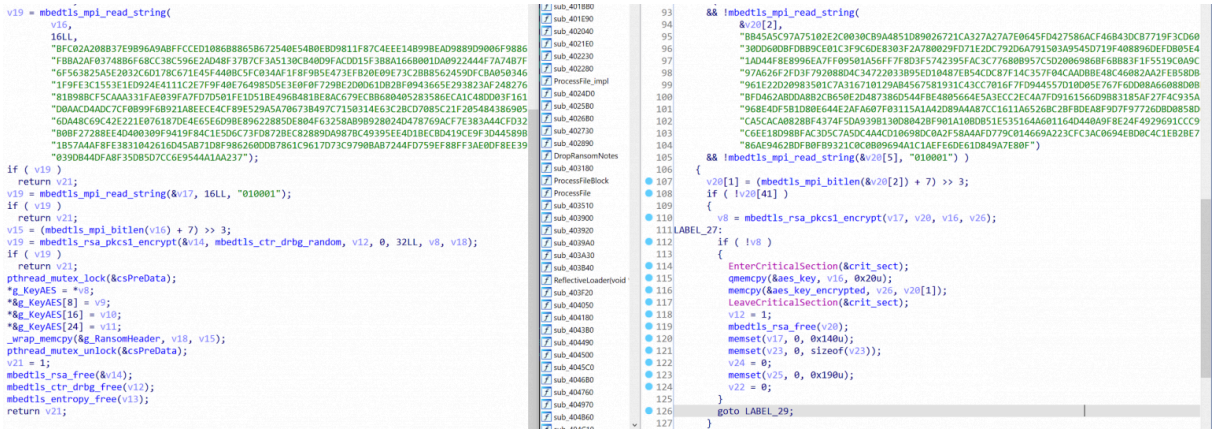
Execution log of the trojan in Kaspersky Linux Sandbox

Similarities with Windows builds of RansomEXX

Despite the fact that previously discovered PE builds of RansomEXX use WinAPI (functions specific to Windows OS), the organization of the Trojan's code and the method of using specific functions from the mbedtls library hint that both ELF and PE may be derived from the same source code.

In the screenshot below, we see a comparison of the procedures that encrypt the AES key. On the left is the ELF sample aa1ddf0c8312349be614ff43e80a262f; on the right is the PE sample fcd21c6fca3b9378961aa1865bee7ecb used in the TxDOT attack.

Despite being built by different compilers with different optimization options and for different platforms, the similarity is quite obvious.



We also observe resemblances in the procedure that encrypts the file content, and in the overall layout of the code.

What's more, the text of the ransom note is also practically the same, with the name of the victim in the title and equivalent phrasing.

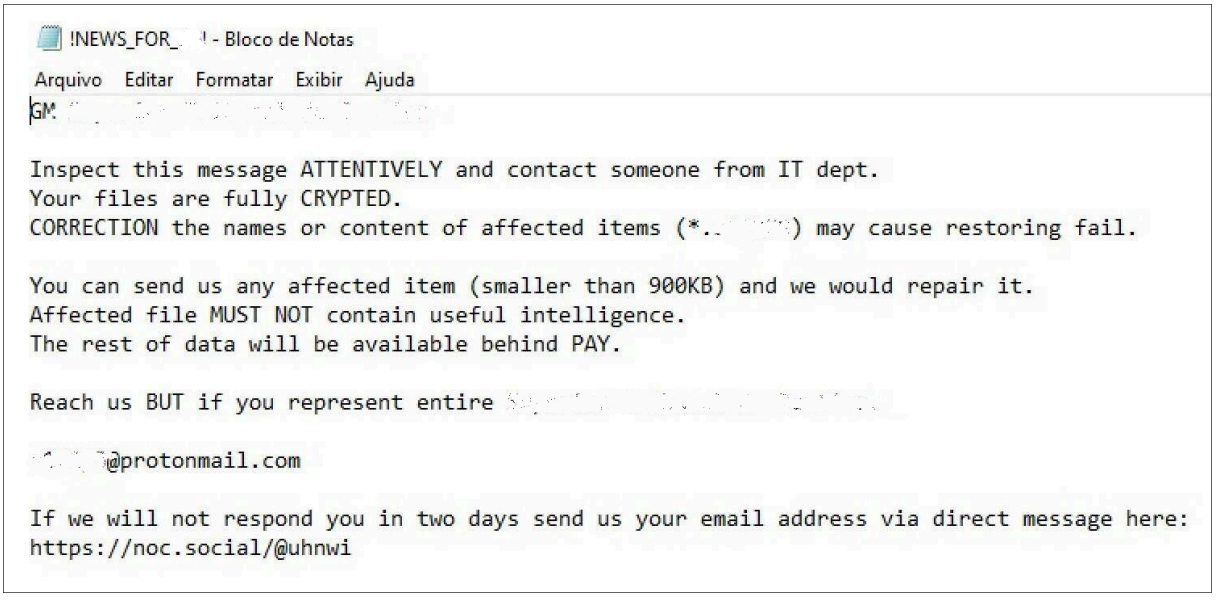
Parallels with a recent attack in Brazil

As reported by the media, one of the country's government institutions has just been attacked by a targeted ransomware Trojan.

Based on the ransom note, which is almost identical to the one in the sample we described, and the news article mentioned above, there is a high probability that the target is the victim of another variant of RansomEXX.

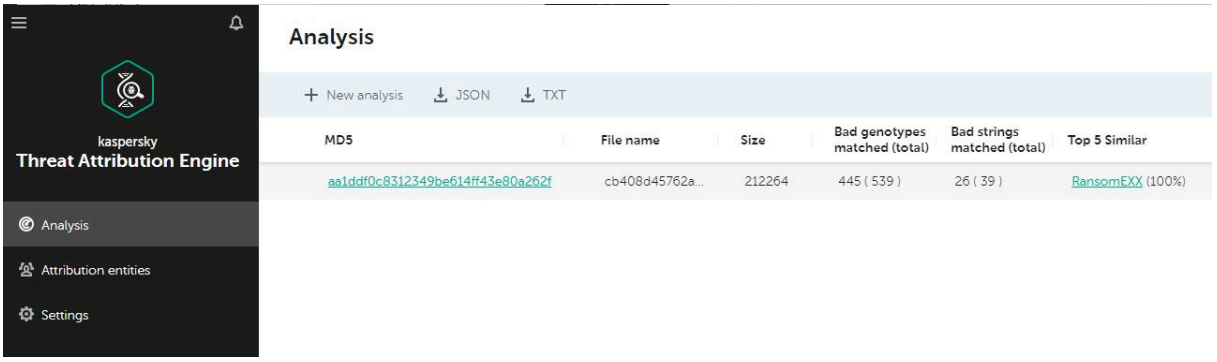
```
db 'Greetings !!!',0Dh,0Ah
                        ; DATA XREF: ReadMeStoreForDir+119↑o
db 0Dh,0Ah
db 'Study this message REGARDFULLY and call administrator from techni'
db 'cal division.',0Dh,0Ah
db 'Yours information is securely ENCRYPTED.',0Dh,0Ah
db 'CHANGING content or names of crypted files (*. ) can make rec'
db 'overing failure.',0Dh,0Ah
db 0Dh,0Ah
db 'You can mail us one crypted document (not bigger than 700KB) and '
db 'we would restore it.',0Dh,0Ah
db 'Encrypted file MUST NOT have rich data.',0Dh,0Ah
db 'All other data will be your behind the PAYMENT.',0Dh,0Ah
db 0Dh,0Ah
db 'Reach us SOLELY if you represent all affected network.',0Dh,0Ah
db 0Dh,0Ah
db '@protonmail.com',0
align 20h
```

Ransom note from the sample aa1ddf0c8312349be614ff43e80a262f



Ransom note from the Bleeping Computer post about the most recent attack in Brazil

Our products protect against this threat and detect it as Trojan-Ransom.Linux.Ransomexx



Kaspersky Threat Attribution Engine identifies Ransomexx malware family

Indicators of compromise

Recent Linux version: aa1ddf0c8312349be614ff43e80a262f
Earlier Windows version: fcd21c6fca3b9378961aa1865bee7ecb

Source: https://securelist.com/ransomexx-trojan-attacks-linux-systems/99279/