KeyPass ransomware

By Orkhan Mamedov

Published: 2018-08-13 · Archived: 2026-04-05 22:02:40 UTC

In the last few days, our anti-ransomware module has been detecting a new variant of malware – KeyPass ransomware. Others in the security community have also noticed that this ransomware began to actively spread in August:

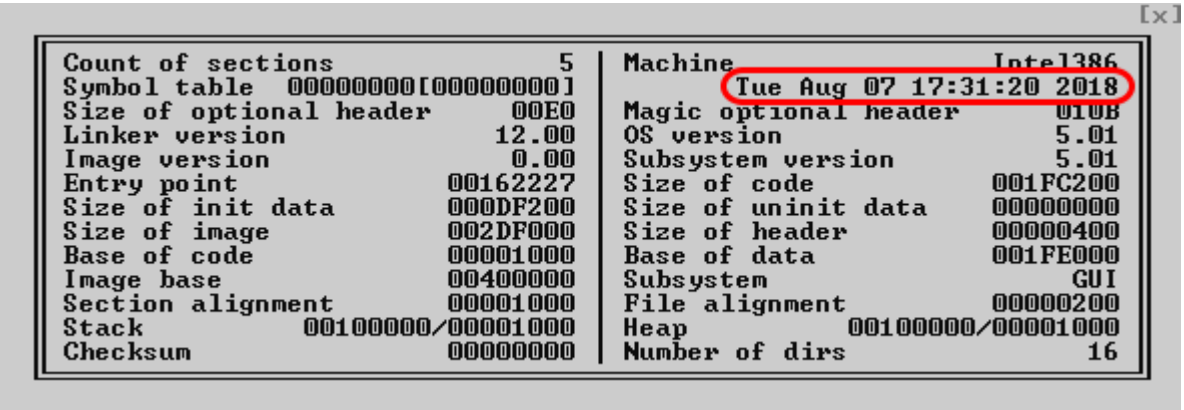
Notification from MalwareHunterTeam

Distribution model

According to our information, the malware is propagated by means of fake installers that download the ransomware module.

Description

The Trojan sample is written in C++ and compiled in MS Visual Studio. It was developed using the libraries MFC, Boost and Crypto++. The PE header contains a recent compilation date.

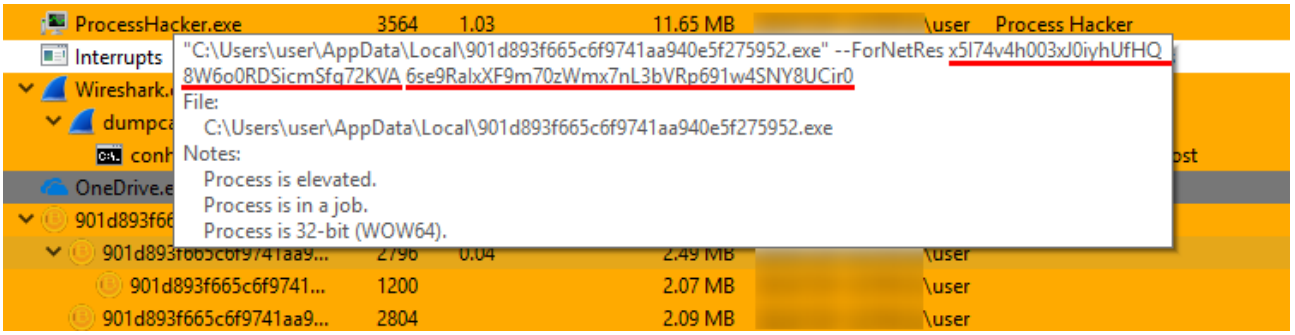


Count of sections	5	Machine	Intel386
Symbol table	00000000[00000000]		Tue Aug 07 17:31:20 2018
Size of optional header	00E0	Magic optional header	010B
Linker version	12.00	OS version	5.01
Image version	0.00	Subsystem version	5.01
Entry point	00162227	Size of code	001FC200
Size of init data	000DF200	Size of uninit data	00000000
Size of image	002DF000	Size of header	00000400
Base of code	00001000	Base of data	001FE000
Image base	00400000	Subsystem	GUI
Section alignment	00001000	File alignment	00000200
Stack	00100000/00001000	Heap	00100000/00001000
Checksum	00000000	Number of dirs	16

PE header with compilation date

When started on the victim's computer, the Trojan copies its executable to %LocalAppData% and launches it. It then deletes itself from the original location.

Following that, it spawns several copies of its own process, passing the encryption key and victim ID as command line arguments.



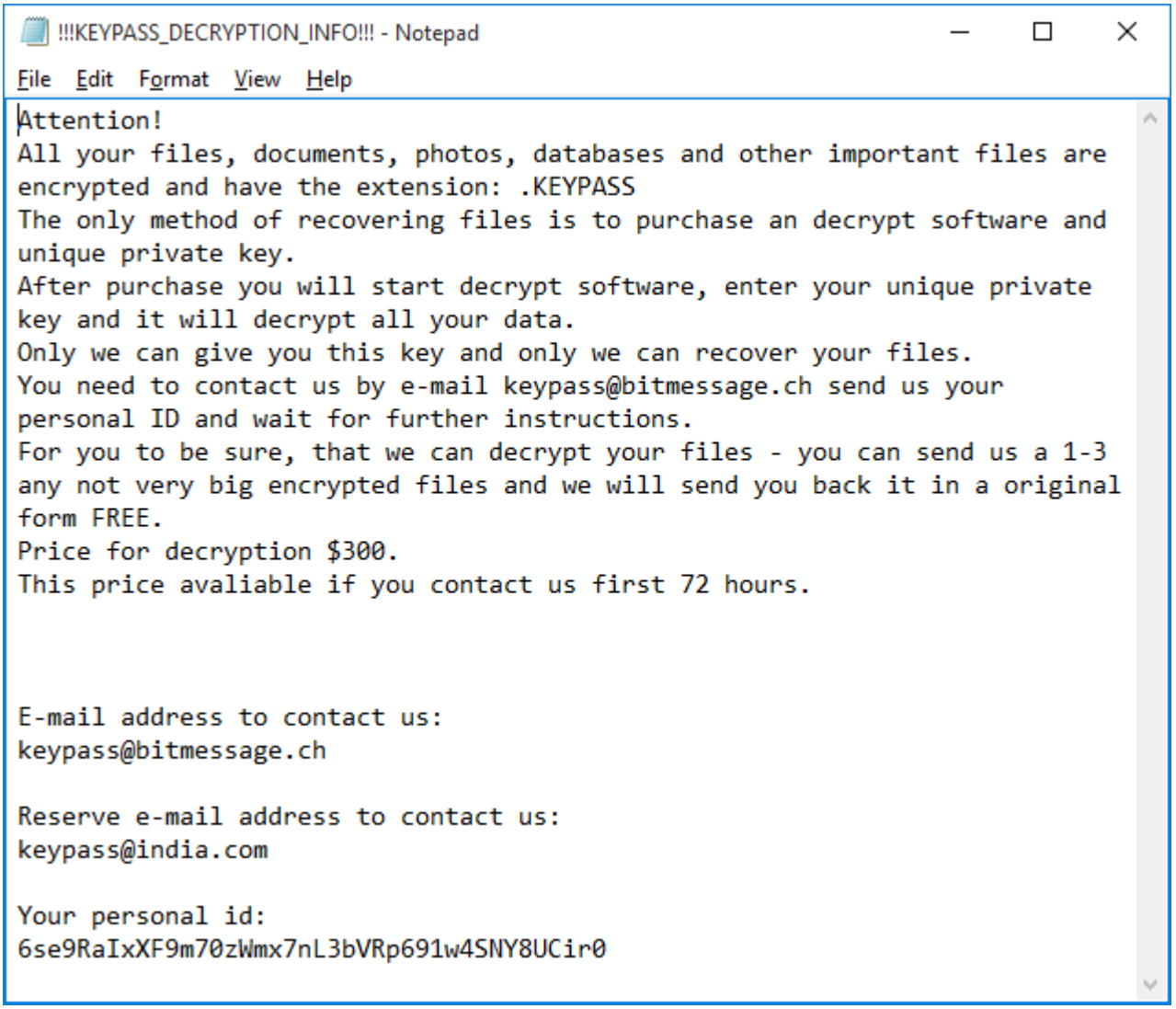
Command line arguments

KeyPass enumerates local drives and network shares accessible from the infected machine and searches for all files, regardless of their extension. It skips files located in a number of directories, the paths to which are hardcoded into the sample.

```
.rdata:00642DE4 43 00 3A 00 5C 00 57 00+                text "UTF-16LE", 'C:\Windows\',0
.rdata:00642DFC 00 00 00 00                             align 10h
.rdata:00642E00                         aCProgramFilesX:                        ; DATA XREF: sub_416030:loc_4161DC↑o
.rdata:00642E00 43 00 3A 00 5C 00 50 00+                text "UTF-16LE", 'C:\Program Files (x86)\Mozilla Firefox\',0
.rdata:00642E50                         aCProgramFilesX_0:                      ; DATA XREF: sub_416030:loc_416223↑o
.rdata:00642E50 43 00 3A 00 5C 00 50 00+                text "UTF-16LE", 'C:\Program Files (x86)\Internet Explorer\',0
.rdata:00642EA4                         aCProgramFilesX_1:                      ; DATA XREF: sub_416030:loc_41626A↑o
.rdata:00642EA4 43 00 3A 00 5C 00 50 00+                text "UTF-16LE", 'C:\Program Files (x86)\Google\',0
.rdata:00642EE2 00 00 00 00 00 00                       align 8
.rdata:00642EE8                         aCProgramFilesM:                        ; DATA XREF: sub_416030:loc_4162B1↑o
.rdata:00642EE8 43 00 3A 00 5C 00 50 00+                text "UTF-16LE", 'C:\Program Files\Mozilla Firefox\',0
.rdata:00642F2C 00 00 00 00                             align 10h
.rdata:00642F30                         aCProgramFilesI:                        ; DATA XREF: sub_416030:loc_4162F8↑o
.rdata:00642F30 43 00 3A 00 5C 00 50 00+                text "UTF-16LE", 'C:\Program Files\Internet Explorer\',0
.rdata:00642F78                         aCProgramFilesG:                        ; DATA XREF: sub_416030:loc_41633F↑o
.rdata:00642F78 43 00 3A 00 5C 00 50 00+                text "UTF-16LE", 'C:\Program Files\Google\',0
.rdata:00642FAA 00 00                                   align 4
.rdata:00642FAC                         aDWindows:                              ; DATA XREF: sub_416030:loc_416386↑o
.rdata:00642FAC 44 00 3A 00 5C 00 57 00+                text "UTF-16LE", 'D:\Windows\',0
.rdata:00642FC4 00 00 00 00                             align 8
.rdata:00642FC8                         aDProgramFilesX:                        ; DATA XREF: sub_416030:loc_4163CD↑o
.rdata:00642FC8 44 00 3A 00 5C 00 50 00+                text "UTF-16LE", 'D:\Program Files (x86)\Mozilla Firefox\',0
.rdata:00643018                         aDProgramFilesX_0:                      ; DATA XREF: sub_416030:loc_416414↑o
.rdata:00643018 44 00 3A 00 5C 00 50 00+                text "UTF-16LE", 'D:\Program Files (x86)\Internet Explorer\',0
.rdata:0064306C                         aDProgramFilesX_1:                      ; DATA XREF: sub_416030:loc_41645B↑o
.rdata:0064306C 44 00 3A 00 5C 00 50 00+                text "UTF-16LE", 'D:\Program Files (x86)\Google\',0
.rdata:006430AA 00 00 00 00 00 00                       align 10h
.rdata:006430B0                         aDProgramFilesM:                        ; DATA XREF: sub_416030:loc_4164A2↑o
.rdata:006430B0 44 00 3A 00 5C 00 50 00+                text "UTF-16LE", 'D:\Program Files\Mozilla Firefox\',0
.rdata:006430F4 00 00 00 00                             align 8
.rdata:006430F8                         aDProgramFilesI:                        ; DATA XREF: sub_416030:loc_4164E9↑o
.rdata:006430F8 44 00 3A 00 5C 00 50 00+                text "UTF-16LE", 'D:\Program Files\Internet Explorer\',0
.rdata:00643140                         aDProgramFilesG:                        ; DATA XREF: sub_416030:loc_416530↑o
.rdata:00643140 44 00 3A 00 5C 00 50 00+                text "UTF-16LE", 'D:\Program Files\Google\',0
```

The list of excluded paths

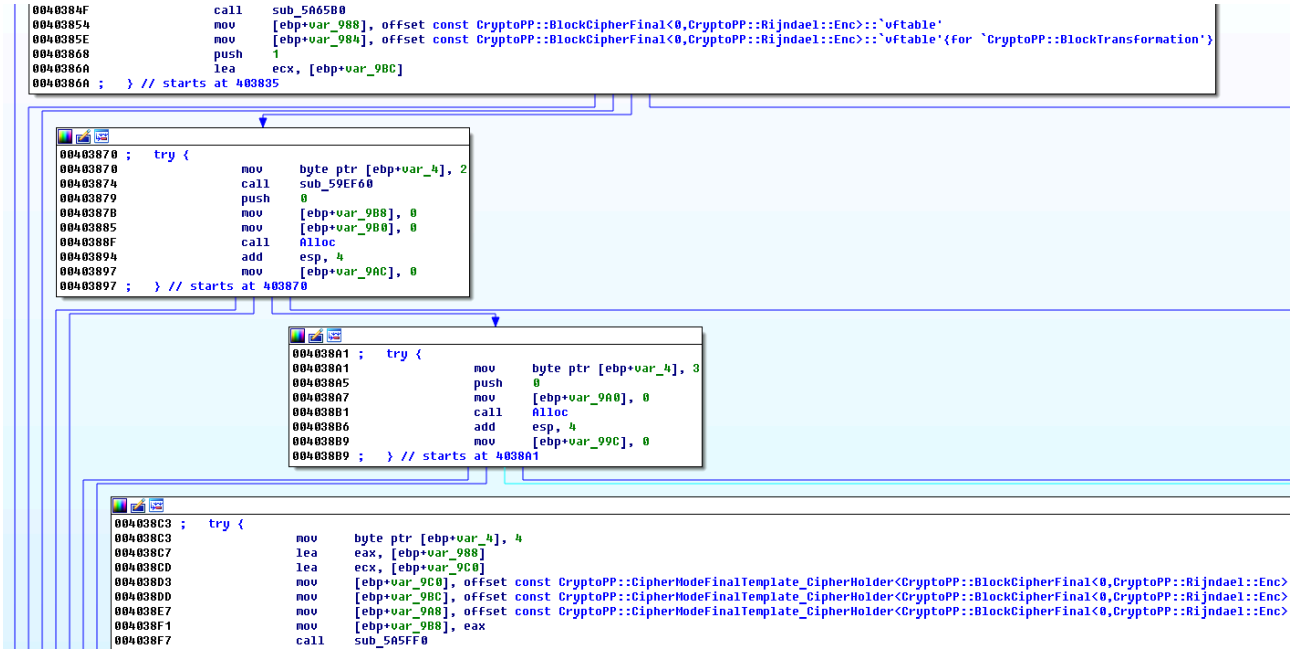
Every encrypted file gets an additional extension: “.KEYPASS” and ransom notes named “”!!!KEYPASS_DECRYPTION_INFO!!!.txt”” are saved in each processed directory.



The ransom note

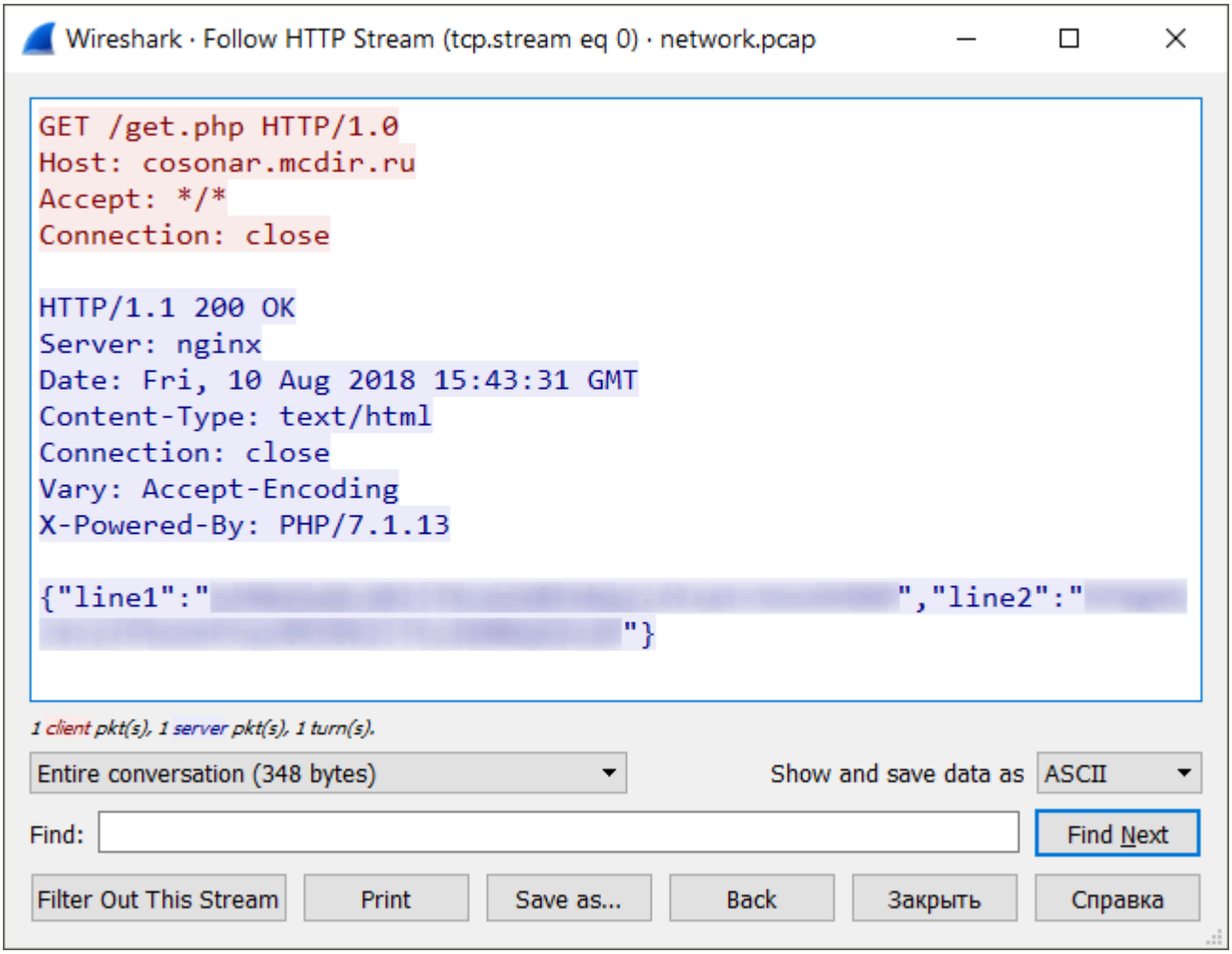
Encryption scheme

The developers of this Trojan implemented a very simplistic scheme. The malware uses the symmetric algorithm AES-256 in CFB mode with zero IV and the same 32-byte key for all files. The Trojan encrypts a maximum of 0x500000 bytes (~5 MB) of data at the beginning of each file.



Part of the procedure that implements data encryption

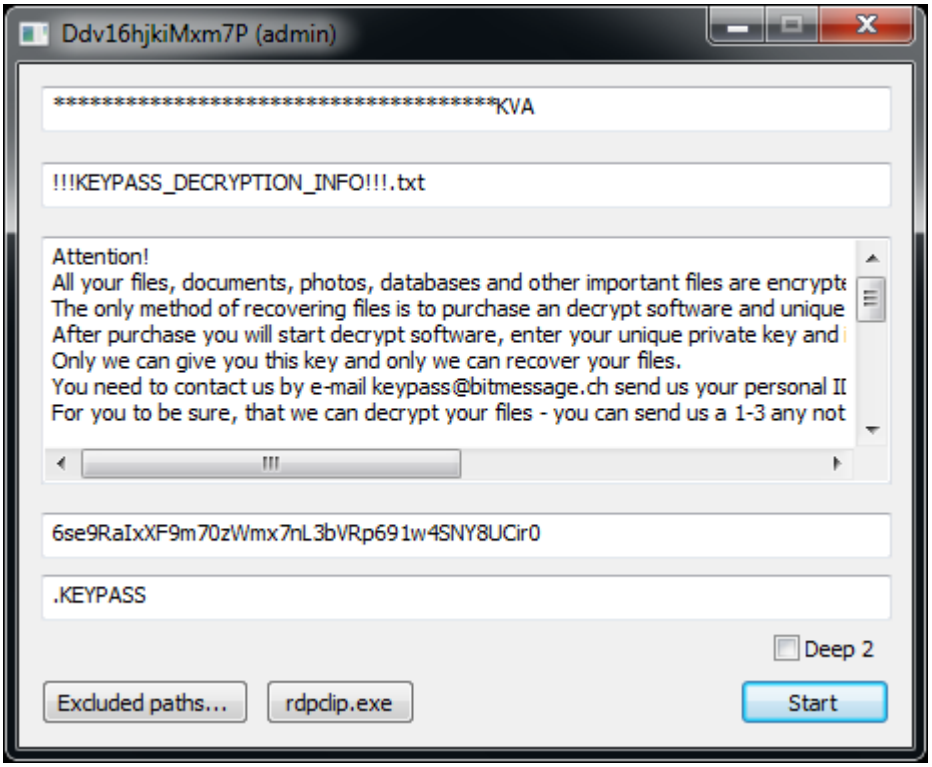
Soon after launch, KeyPass connects to its command and control (C&C) server and receives the encryption key and the infection ID for the current victim. The data is transferred over plain HTTP in the form of JSON.



If the C&C is inaccessible (e.g. if the infected machine is not connected to the internet or the server is down), the Trojan uses a hardcoded key and ID, which means that in the case of offline encryption the decryption of the victim's files will be trivial.

GUI

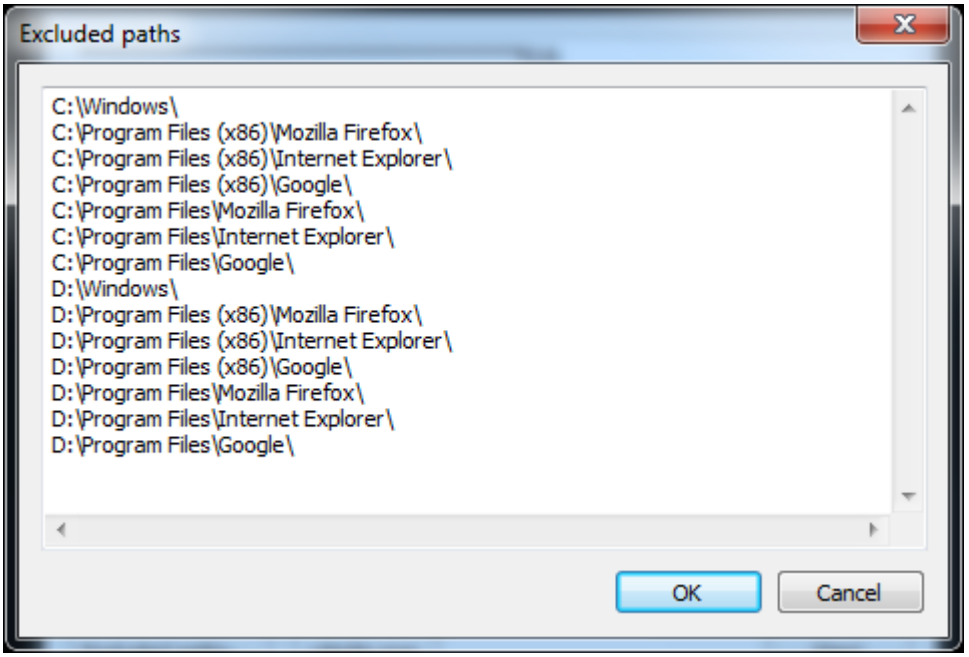
From our point of view, the most interesting feature of the KeyPass Trojan is the ability to take 'manual control'. The Trojan contains a form that is hidden by default, but which can be shown after pressing a special button on the keyboard. This capability might be an indication that the criminals behind the Trojan intend to use it in manual attacks.



GUI of the trojan

This form allows the attacker to customize the encryption process by changing such parameters as:

- encryption key
- name of ransom note
- text of ransom note
- victim ID
- extension of the encrypted files
- list of paths to be excluded from the encryption



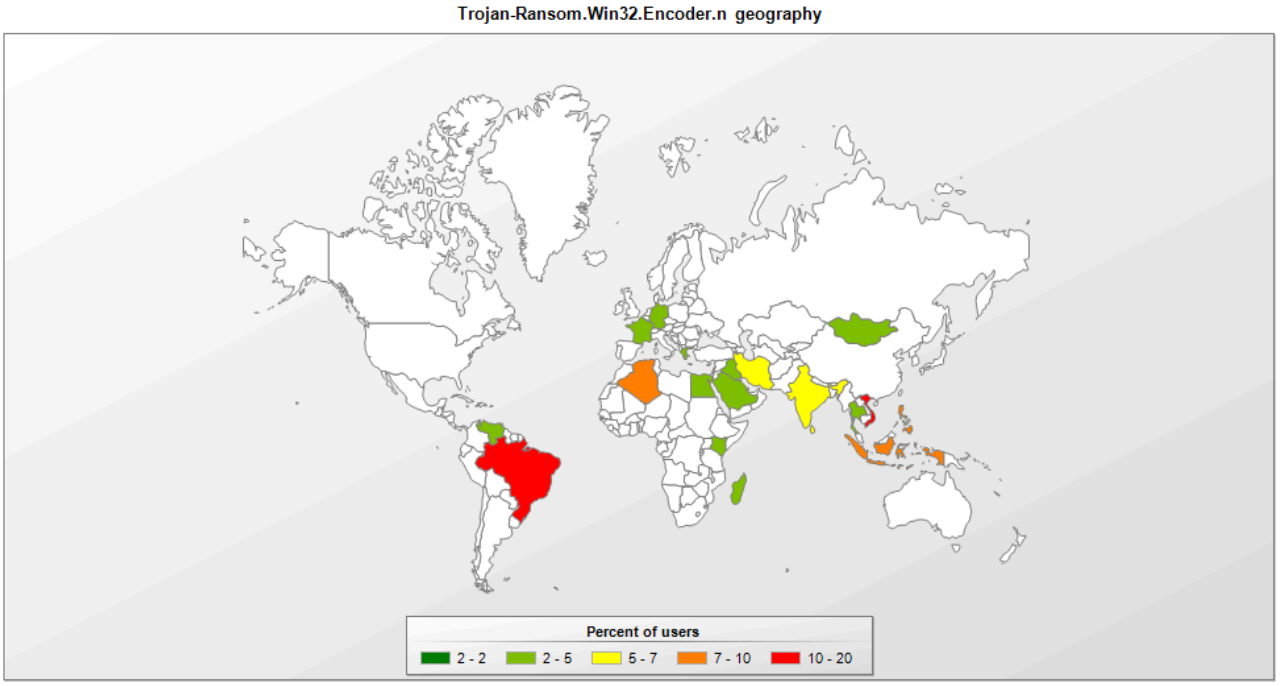
Paths excluded from encryption by default

```
LRESULT __stdcall sub_413780(int code, WPARAM wParam, LPARAM lParam)
{
  int v3; // eax
  int v4; // eax
  HWND hWnd; // esi

  if ( !code && (wParam == WM_KEYDOWN || wParam == WM_SYSKEYDOWN) && *(_DWORD *)lParam == VK_F8 )
  {
    v3 = sub_453BED();
    if ( v3 && (v4 = (*(int (__thiscall **)(int))(*(_DWORD *)v3 + 116))(v3)) != 0 )
      hWnd = *(HWND *)(v4 + 32);
    else
      hWnd = 0;
    if ( IsWindowVisible(hWnd) )
    {
      ShowWindow(hWnd, SW_HIDE);
    }
    else
    {
      ShowWindow(hWnd, SW_SHOW);
      SetForegroundWindow(hWnd);
    }
  }
  return CallNextHookEx(hhk, code, wParam, lParam);
}
```

Pseudocode of the procedure that shows the GUI by a keypress

Geography



IOC

901d893f665c6f9741aa940e5f275952 – Trojan-Ransom.Win32.Encoder.n

hxxp://cosonar.mcdir.ru/get.php

Source: https://securelist.com/keypass-ransomware/87412/