Locky: the encryptor taking the world by storm

By Fedor Sinitsyn

Published: 2016-04-06 · Archived: 2026-04-05 16:47:46 UTC

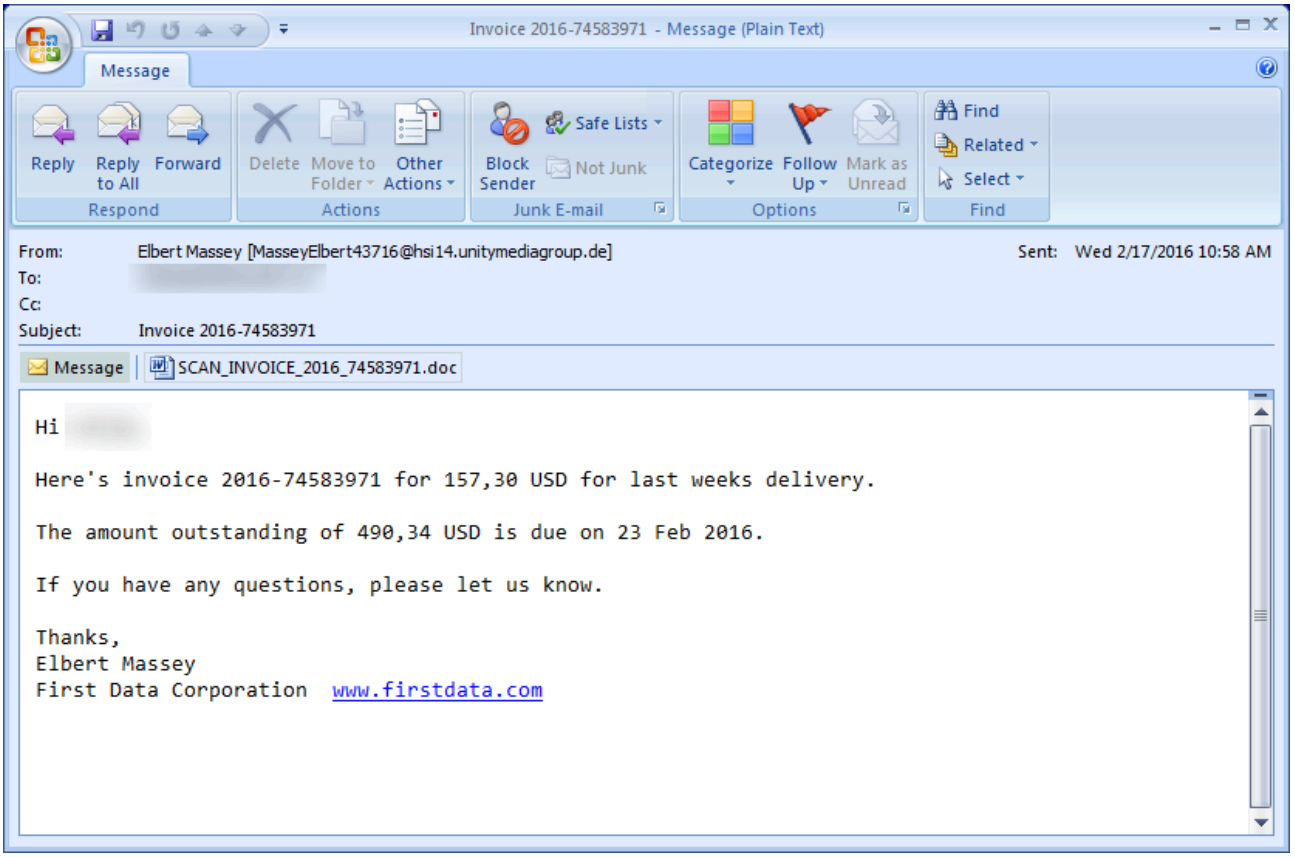
In February 2016, the Internet was shaken by an epidemic caused by the new ransomware Trojan Locky (detected by Kaspersky Lab products as Trojan-Ransom.Win32.Locky). The Trojan has been actively propagating up to the present day. Kaspersky Lab products have reported attempts to infect users with the Trojan in 114 countries around the world.

Analysis of the samples has shown that this Trojan is a brand new ransomware threat, written from scratch. So, what is Locky, and how can we protect against it?

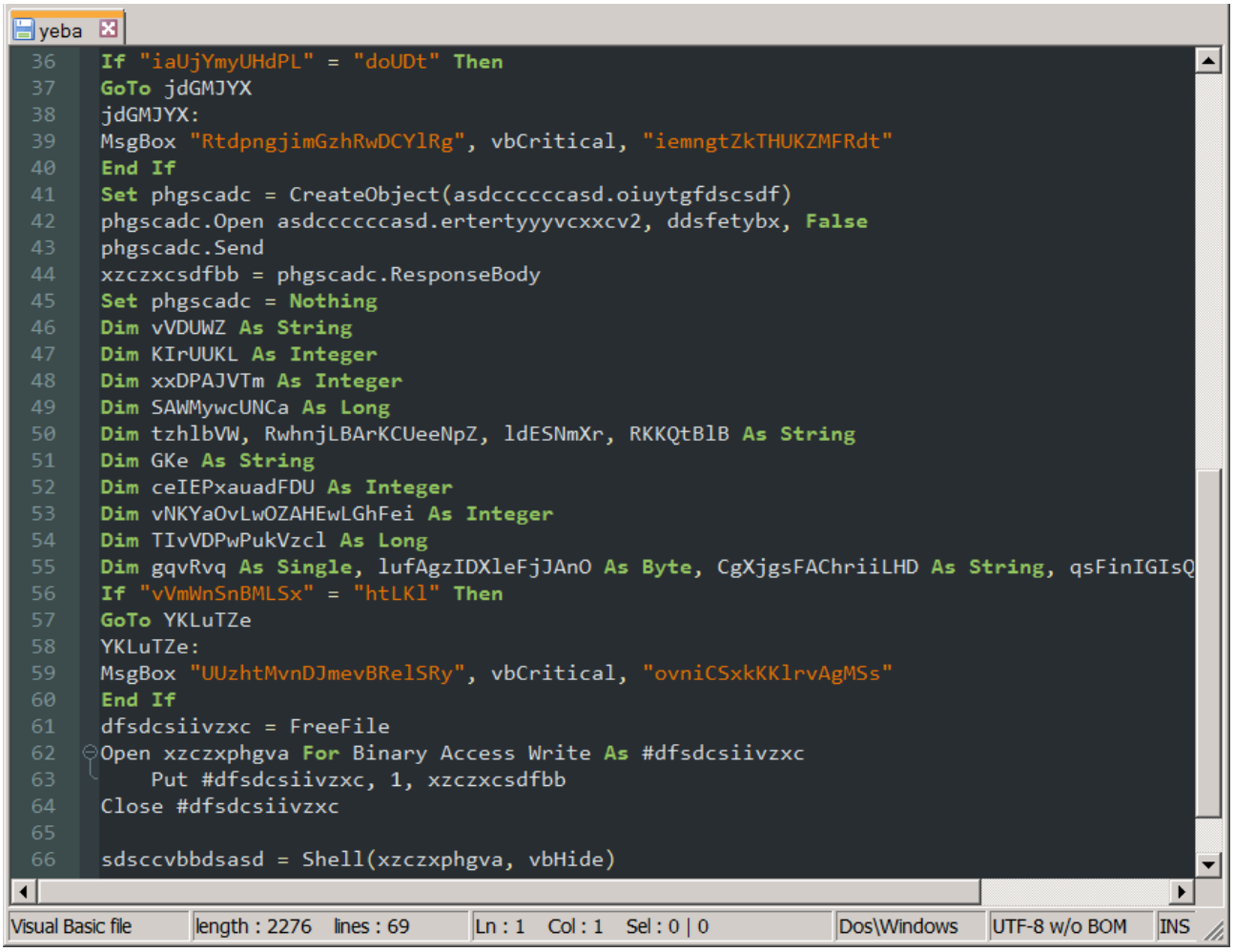
Propagation

In order to spread the Trojan, cybercriminals sent out mass mailings with malicious loaders attached to spam messages.

Initially, the malicious spam messages contained an attached DOC file with a macro that downloaded the Locky Trojan from a remote server and executed it.



An early-stage spam message with a malicious document attached

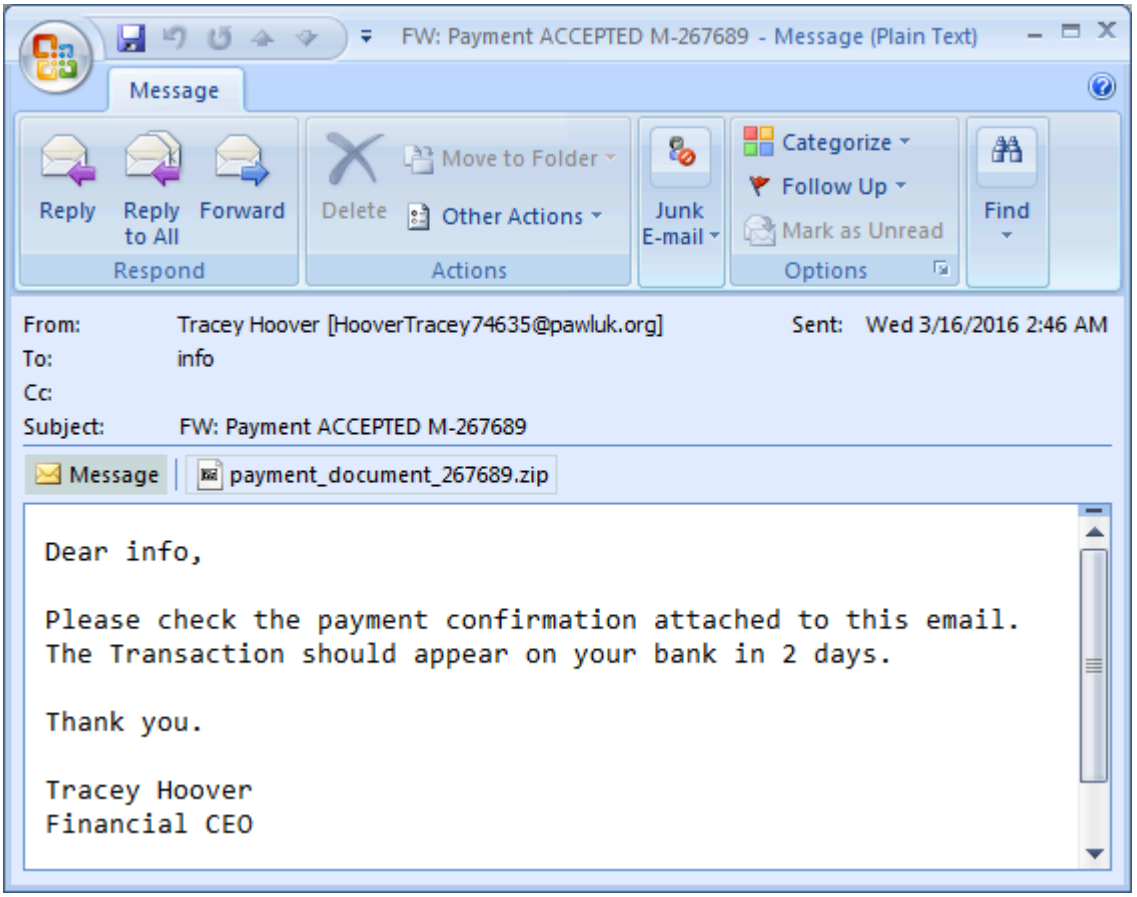


A fragment of the malicious macro

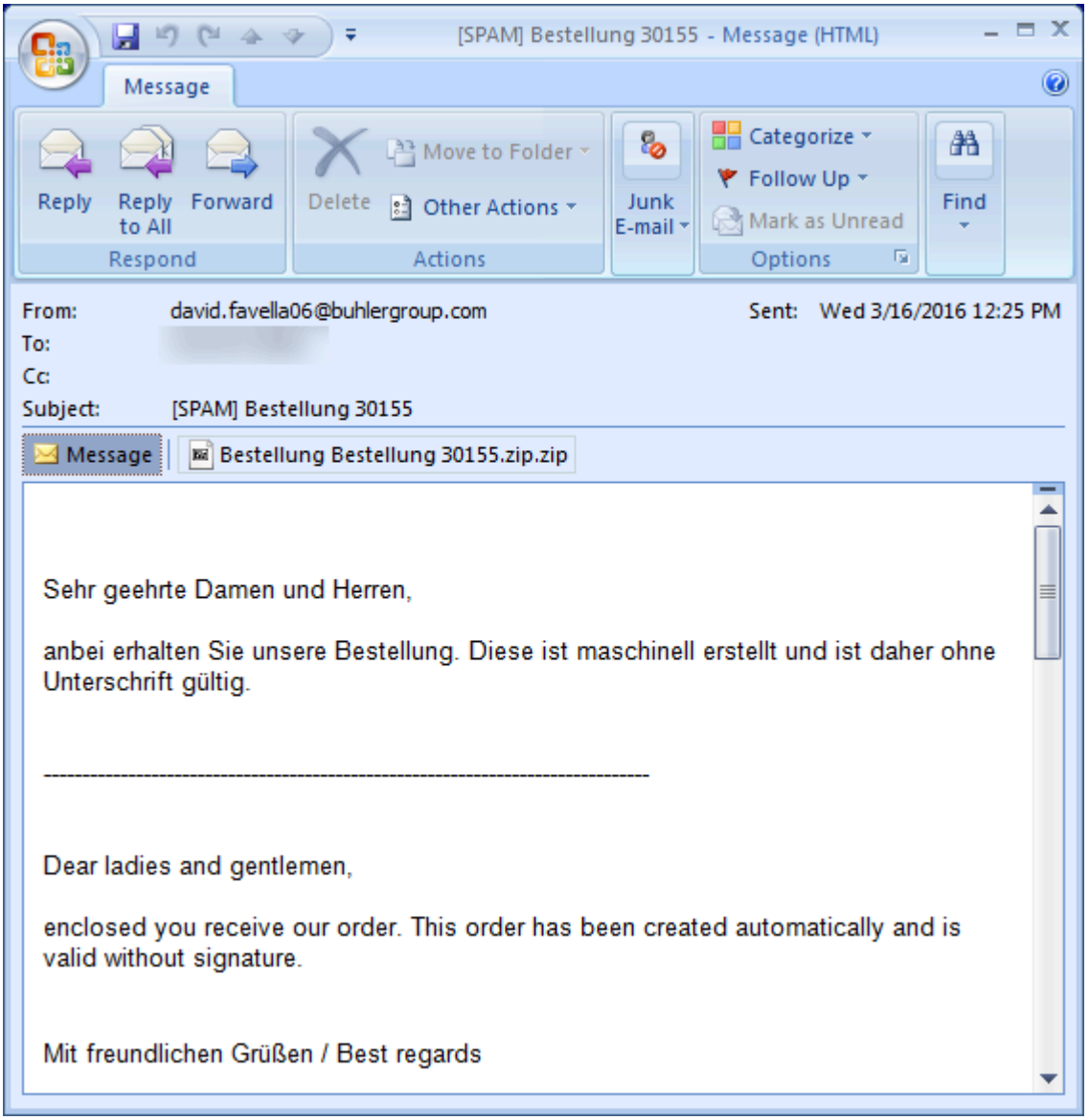
Kaspersky Lab products detect files with malicious macros as Trojan-Downloader.MSWord.Agent and HEUR:Trojan-Downloader.Script.Generic.

We should note that in modern versions of Microsoft Office, automatic execution of macros is disabled for security reasons. However, practice shows that users often enable macros manually, even in documents from unknown sources, which may lead to some damaging consequences.

At the time of writing, the malicious spam is still being sent, but instead of the DOC files being attached there are now ZIP archives containing one or more obfuscated scripts in JavaScript. The messages are mostly in English, though some bilingual variants have appeared.

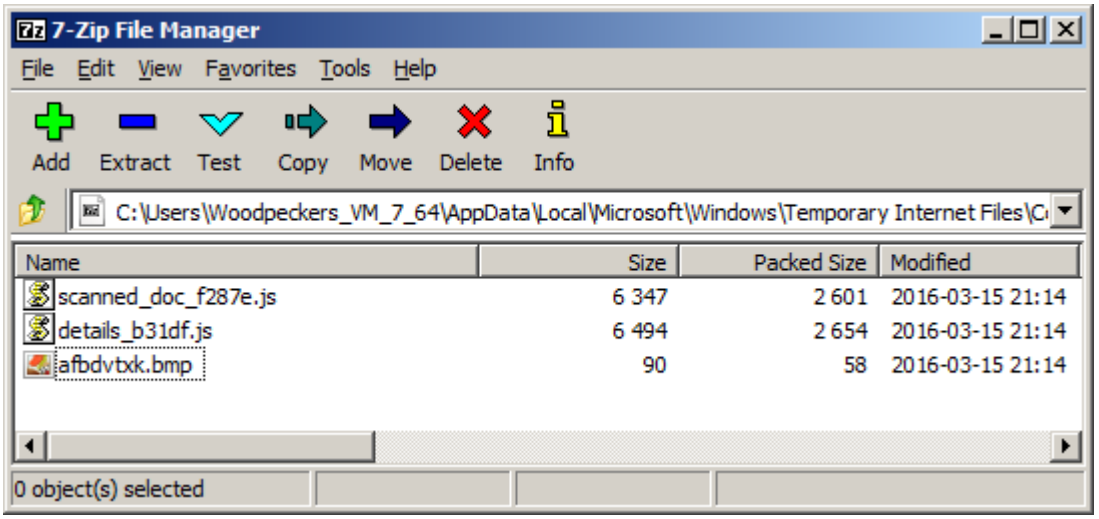


Spam message in English with the archive attached

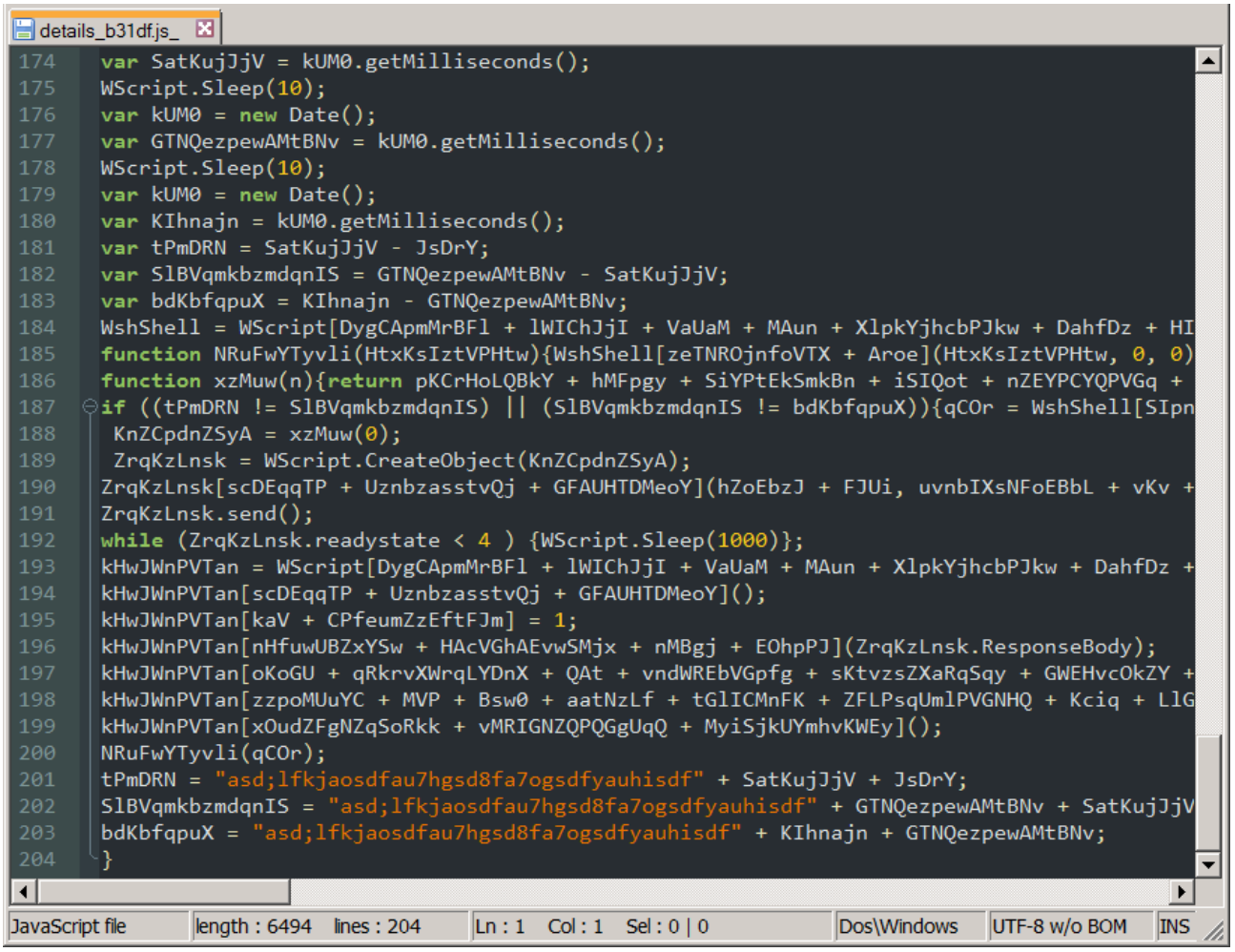


Message in German and English with the archive attached

The user is prompted to manually launch the scripts.



Contents of the archive attached to the message



Fragment of the archived script

When launched, the script downloads the Locky Trojan from a remote server and launches it.

Kaspersky Lab products detect these script loaders as Trojan-Downloader.JS.Agent and HEUR:Trojan-Downloader.Script.Generic.

Geography of attacks

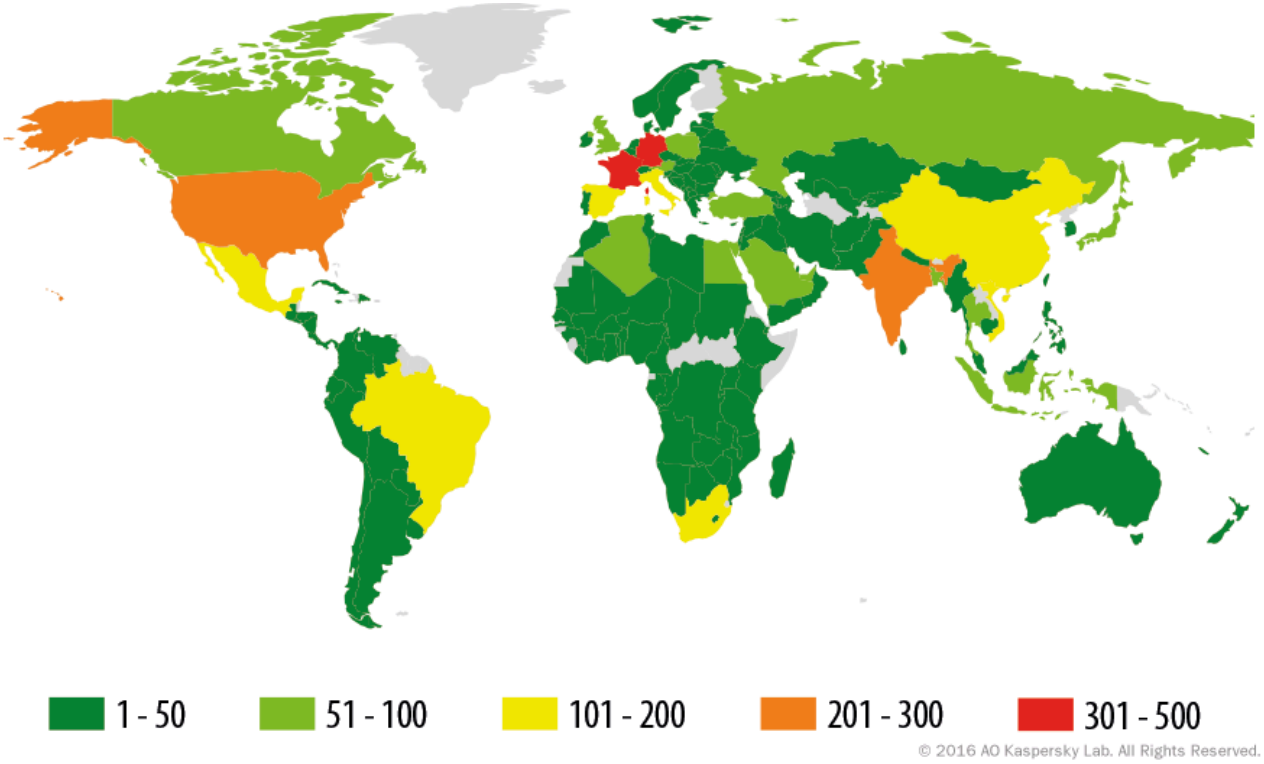
Kaspersky Security Network has reported Locky attacks in 114 countries.

TOP 10 countries

Country	Number of users attacked
France	469
Germany	340
India	267

USA	224
Republic of South Africa	182
Italy	171
Mexico	159
Brazil	156
China	126
Vietnam	107

We should note that these statistics only include cases where the actual Trojan was detected, and does not include early-stage detections reported as malicious spam or malicious downloaders.



The geography of Trojan-Ransom.Win32.Locky attacks (number of attacked users)

As we can see, the Trojan carries out attacks in practically all regions of the world. We can assume which countries the cybercriminals see as their main targets based on the list of languages used on the ransom payment webpage (see details below).

How it works

The Locky Trojan is an executable file, about 100 kb in size. It is written in C++ using STL, and is compiled in Microsoft Visual Studio. When launching, it copies itself to %TEMP%\svchost.exe and deletes the NTFS data stream Zone.Identifier from its copy – this is done to ensure that when the file is launched, Windows does not

display a notification saying that the file has been downloaded from the Internet and may be potentially dangerous. The Trojan then launches from %TEMP%.

Once launched, the Trojan checks for the presence and the contents of the below registry keys.

Path	Type	Value
HKEY_CURRENT_USER\Software\Locky\id	REG_SZ	Infection ID
HKEY_CURRENT_USER\Software\Locky\pubkey	REG_BINARY	Public RSA key in MSBLOB format
HKEY_CURRENT_USER\Software\Locky\paytext	REG_BINARY	Text shown to the victim
HKEY_CURRENT_USER\Software\Locky\completed	REG_DWORD	Status (whether encryption is completed)

If data already exists in the registry keys (this is the case if the Trojan has launched before, but its previous session aborted for some reason), Locky reads that data and continues with the infection process.

If launched for the first time, the Trojan performs the following actions:

1. 1 Contacts C&C and reports infection;
2. 2 Receives a public RSA-2048 key and infection ID from C&C, saves them in the registry;
3. 3 Sends information about the language of the infected operating system, receives the cybercriminals' ransom demand text that will be shown to the victim, saves the text in the registry;
4. 4 Searches for files with specific extensions on local disk drives, encrypts them;
5. 5 Deletes shadow copies of files;
6. 6 Registers itself for autostart (HKEY_CURRENT_USER\Software\Microsoft\Windows\CurrentVersion\Run);
7. 7 Searches for and encrypts files with specific extensions on network drives and on network file resources with no assigned drive letter;
8. 8 Displays the cybercriminals' ransom demands to the victim;
9. 9 Terminates its process and removes itself.

```
.text:00405F5B                          ; std::basic_string *__usercall GetLanguage@<eax>(std::basic_string *langName@<esi>)
.text:00405F5B                          GetLanguage     proc near               ; CODE XREF: ReportInstall+308↑p
.text:00405F5B                                                                  ; WinMain(x,x,x,x)+51E↑p
.text:00405F5B
.text:00405F5B                          var_25          = byte ptr -25h
.text:00405F5B                          LCData          = byte ptr -24h
.text:00405F5B                          var_4           = dword ptr -4
.text:00405F5B
.text:00405F5B 000 55                                   push    ebp
.text:00405F5C 004 8B EC                                mov     ebp, esp
.text:00405F5E 004 83 EC 24                             sub     esp, 24h
.text:00405F61 028 53                                   push    ebx
.text:00405F62 02C 33 DB                                xor     ebx, ebx
.text:00405F64 02C 89 5D FC                             mov     [ebp+var_4], ebx
.text:00405F67 02C FF 15 EC 00 41 00                    call    ds:GetUserDefaultUILanguage
.text:00405F6D 02C 6A 20                                push    20h             ; cchData
.text:00405F6F 030 8D 4D DC                             lea     ecx, [ebp+LCData]
.text:00405F72 030 51                                   push    ecx             ; lpLCData
.text:00405F73 034 0F B7 C0                             movzx   eax, ax
.text:00405F76 034 6A 59                                push    LOCALE_SISO639LANGNAME ; LCType
.text:00405F78 038 50                                   push    eax             ; Locale
.text:00405F79 03C FF 15 E8 00 41 00                    call    ds:GetLocaleInfoA
.text:00405F7F 02C C7 46 14 0F 00 00 00                 mov     dword ptr [esi+14h], 0Fh
.text:00405F86 02C 89 5E 10                             mov     [esi+10h], ebx
.text:00405F89 02C 88 1E                                mov     [esi], bl
.text:00405F8B 02C 3B C3                                cmp     eax, ebx
.text:00405F8D 02C 7F 4F                                jg      short loc_405FDE
.text:00405F8F 02C 57                                   push    edi
.text:00405F90 030 BF F9 2B 41 00                       mov     edi, offset unk_412BF9
.text:00405F95 030 57                                   push    edi
.text:00405F96 034 8B C6                                mov     eax, esi
.text:00405F98 034 E8 65 F6 FF FF                       call    std__basic_string___Inside_0
.text:00405F9D 030 84 C0                                test    al, al
.text:00405F9F 030 74 1B                                jz      short loc_405FBC
.text:00405FA1 030 83 7E 14 10                          cmp     dword ptr [esi+14h], 10h
.text:00405FA5 030 72 04                                jb      short loc_405FAB
.text:00405FA7 030 8B 06                                mov     eax, [esi]
.text:00405FA9 030 EB 02                                jmp     short loc_405FAD
.text:00405FAB                          ; ---------------------------------------------------------------------------
```

Fragment of code that determines the language of the operating system

File encryption

The Trojan searches for files matching a given list of extensions. Then, these files are encrypted as described below.

File encryption

The Trojan searches for files matching a given list of extensions. Then, these files are encrypted as described below.

List of file extensions that are subject to encryption

For each file that matches an extension on the list, the Trojan generates a new 128-bit key and encrypts the file's contents with the algorithm AES-128 in CTR mode. The encrypted file is given the name <16 HEX characters as ID><16 random HEX characters>.locky. Then the following structure is added to the end of the file:

```
00000000
00000000 file_data       struc ; (sizeof=0x344, mappedto_99) ; XREF: ProcessFile/r
00000000 start_marker    dd ?                    ; XREF: ProcessFile+2C/w
00000004 id              db 16 dup(?)            ; XREF: ProcessFile+43/o
00000014 aes_key         db 256 dup(?)           ; XREF: ProcessFile:loc_401862/o
00000014                                         ; ProcessFile+3B8/o ...
00000114 name_marker     dd ?                    ; XREF: ProcessFile+36/w
00000114                                         ; ProcessFile+449/o
00000118 orig_name       db 520 dup(?)           ; XREF: ProcessFile+9B/o
00000320 attr            WIN32_FILE_ATTRIBUTE_DATA ? ; XREF: ProcessFile:loc_4015B1/o
00000320                                         ; ProcessFile:loc_4015E3/r ...
00000344 file_data       ends
```

Structure appended by the Trojan to the end of an encrypted file

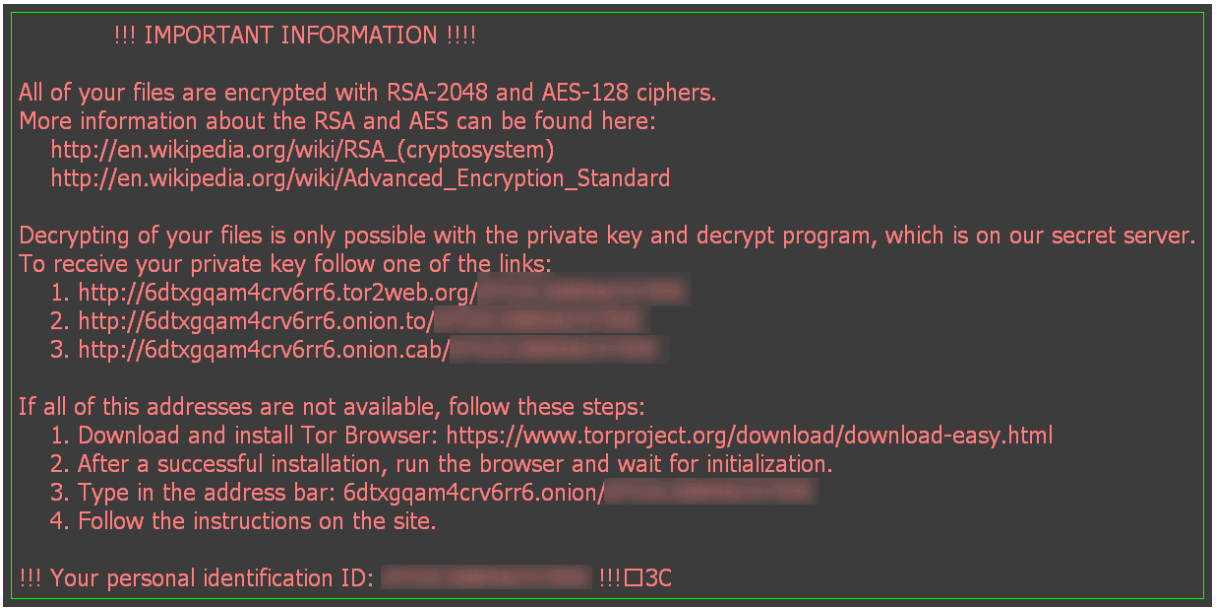
In C language syntax, this structure may be described as follows:

```
struct file_data

{

uint32_t start_marker;        //Structure start marker = 0x8956FE93

char id[16];                  //Infection ID

uint8_t aes_key[256];         //AES key encrypted with RSA-2048

uint32_t name_marker;         //Name start marker encrypted with AES (= 0xD41BA12A after decryption)

uint8_t orig_name[520];       //Original file name encrypted with AES

WIN32_FILE_ATTRIBUTE_DATA attr; //Original file attributes encrypted with AES

};
```

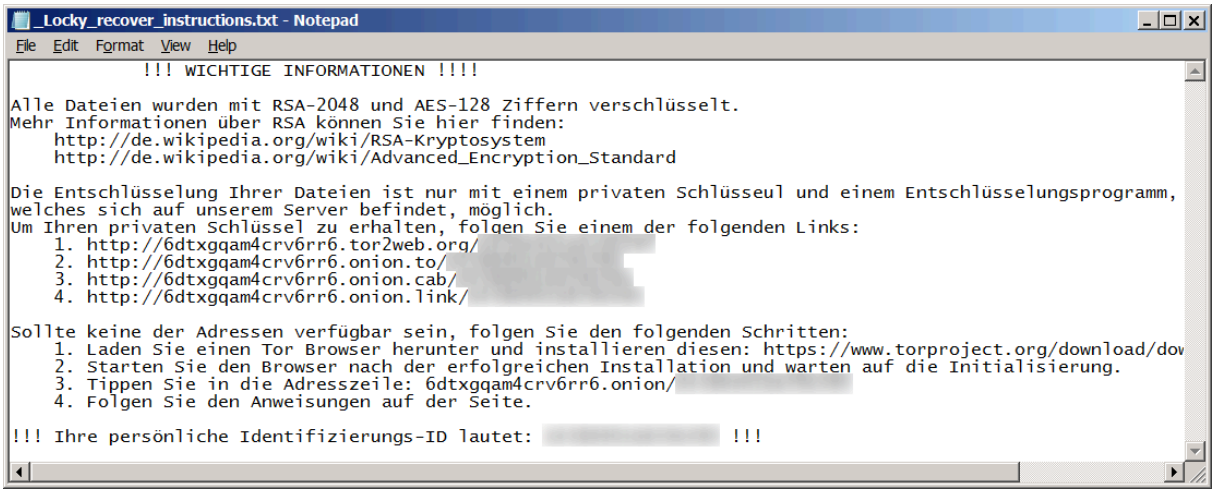
Appended structure described in C language syntax

Ransom demands

After encrypting the user's files, the Trojan displays the following message with the cybercriminals' ransom demands.



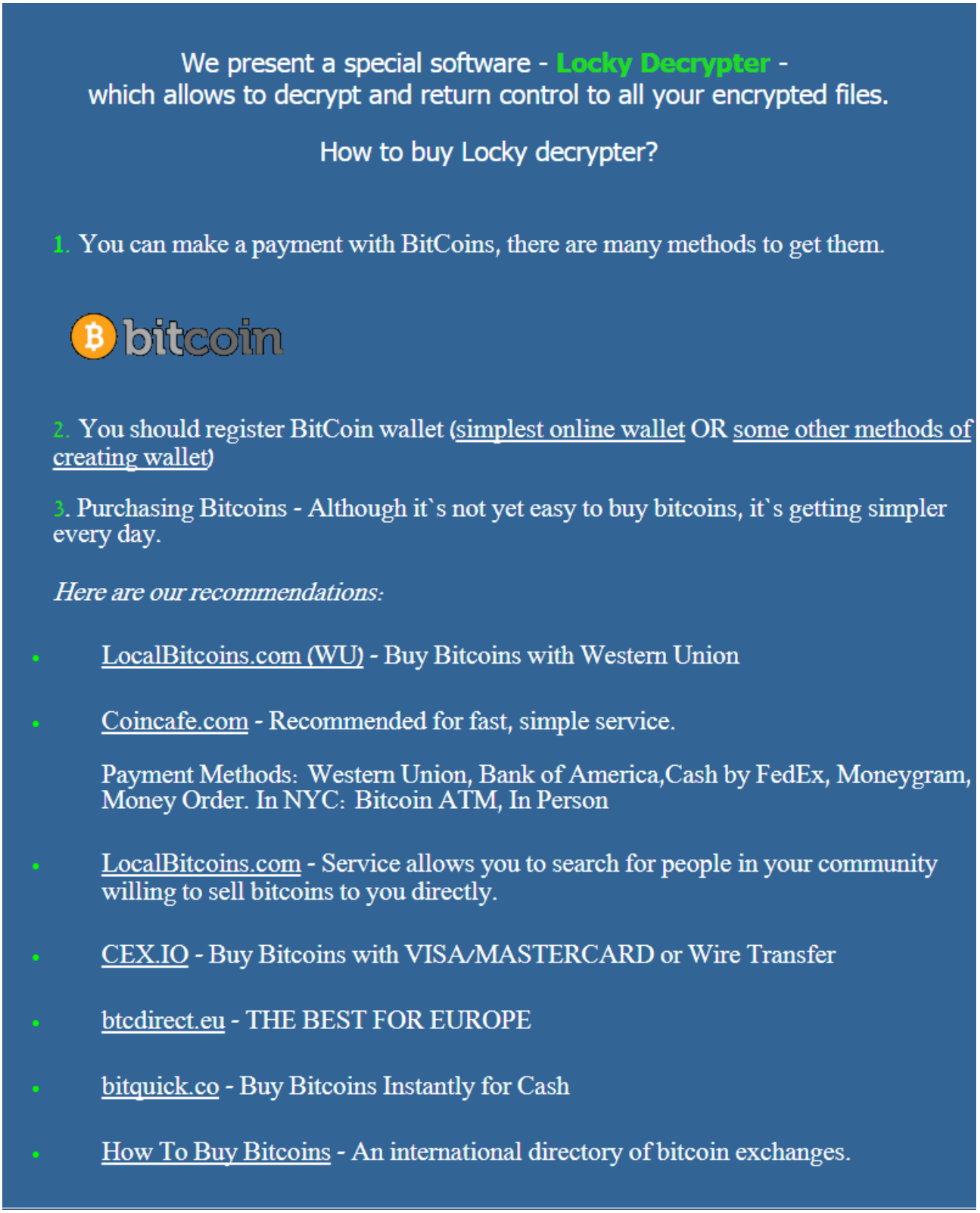
Ransom demand in English



Ransom demand in German

The ransom message contains the address of the cybercriminals' 'secret server' where they placed information about the ransom they demand for the decryption program. All four links in the message lead to the same website in the Tor network.

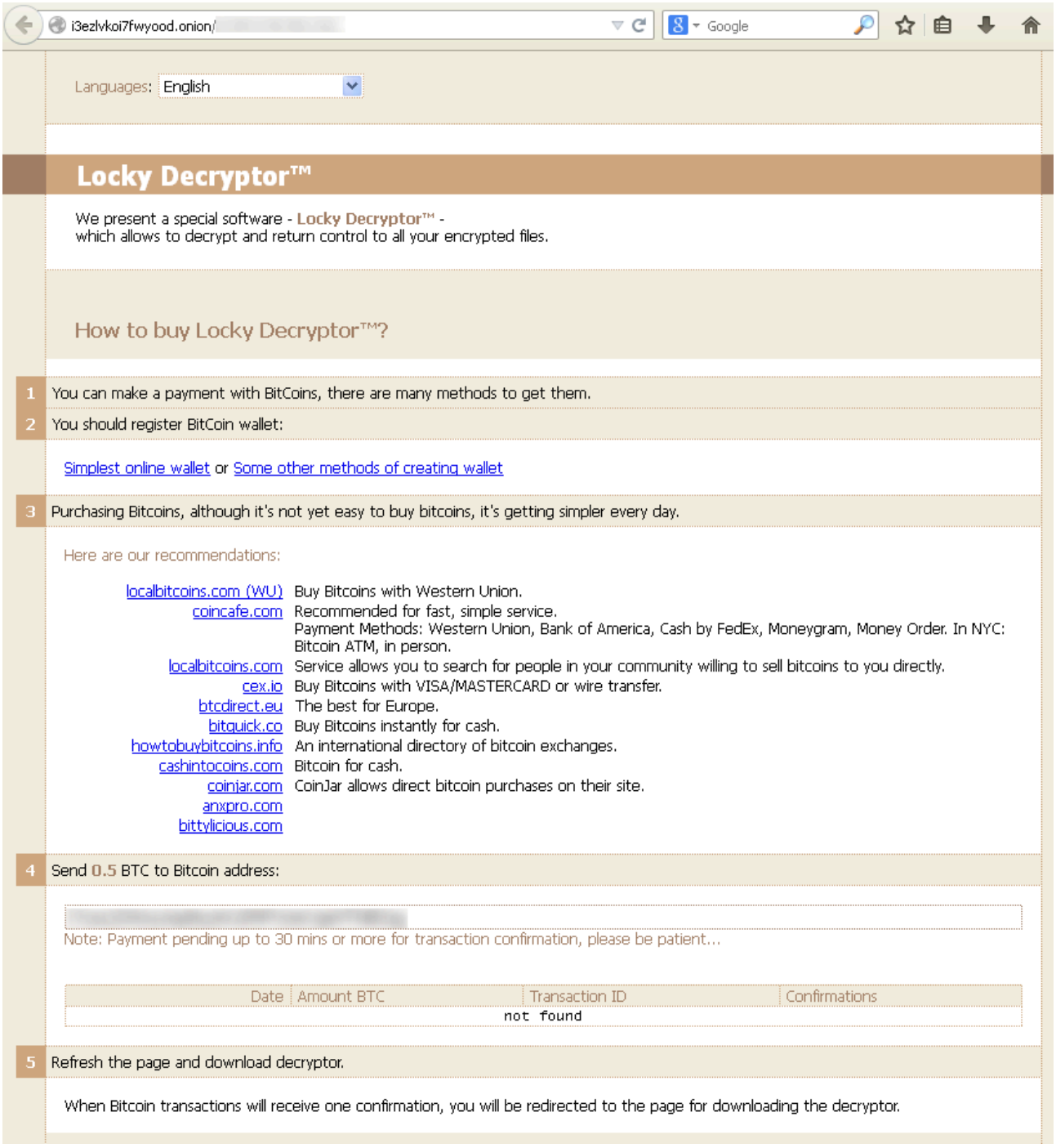
During the early spamming campaigns, the ransom payment page looked like this:



Early version of Locky's ransom demand page

On this page, the cybercriminals suggested that the victims pay in bitcoins to decrypt the affected files on their computer. They also gave recommendations about where and how to get the cryptocurrency.

The contents and the design of the page changed with time. Today, the page is available in more than 20 languages (that can be selected from a dropdown list), and looks like this:



Latest version of Locky's ransom payment page

If we look at the page's source code, we will see a complete list of supported languages. The cybercriminals obviously see the corresponding countries as the main targets for this ransomware Trojan. Interestingly, Russian and other CIS languages are not on the list. For some reason the cybercriminals are not that keen on targeting users in countries where those languages are spoken – something that KSN statistics confirm.

```
</p><form action="/" method="get">
    <font id="brown">Languages</font>:
    <select name="lang" onchange="this.form.submit()">
        <option value="bg">Български</option>
        <option value="ca">Català</option>
        <option value="cs">Čeština</option>
        <option value="da">Dansk</option>
        <option value="de">Deutsch</option>
        <option value="el">Ελληνικά</option>
        <option value="en" selected="selected">English</option>
        <option value="es">Español</option>
        <option value="fi">Suomi</option>
        <option value="fr">Français</option>
        <option value="hi">हिन्दी</option>
        <option value="hr">Hrvatski</option>
        <option value="hu">Magyar</option>
        <option value="it">Italiano</option>
        <option value="ja">日本語</option>
        <option value="ko">한국어</option>
        <option value="ms">Bahasa Melayu</option>
        <option value="nl">Nederlands</option>
        <option value="no">Norsk bokmål</option>
        <option value="pl">Polski</option>
        <option value="pt">Português</option>
        <option value="sk">Slovenčina</option>
        <option value="sr">Српски</option>
        <option value="sv">Svenska</option>
        <option value="tr">Türkçe</option>
        <option value="zh">中文</option>
    </select>
```

List of languages supported on Locky ransom payment page

Communication with C&C

The Trojan's code contains between one and three C&C IP addresses. On top of that, the code contains an algorithm generating new C&C addresses (DGA, domain generation algorithm) depending on the current day, month and year. With this algorithm, six C&C addresses are generated each day. The pseudo-code to illustrate the DGA Locky algorithm is highlighted in the screenshot below.

```
GetSystemTime(&SystemTime);
n1 = __ROR4__(0xB11924E1 * (SystemTime.wYear + 0x1BF5), 5);
n2 = __ROR4__(0xB11924E1 * (n1 + ((unsigned int)SystemTime.wDay >> 1) + 0x27100001), 5);
n3 = __ROR4__(0xB11924E1 * (n2 + SystemTime.wMonth + 0x2709A354), 5);
n4 = __ROL4__(seed % 6, 21);
n5 = __ROR4__(0xB11924E1 * (n3 + n4 + 0x27100001), 5);
n6 = n5 + 0x27100001;
n7 = (n5 + 0x27100001) % 11u + 5;
std::basic_string::f_17((n5 + 0x27100001) % 11u + 8, &str);
v22 = 0;
if ( n7 )
{
  do
  {
    n8 = __ROL4__(n6, i);
    v10 = (std::basic_string *)str._Bx._Ptr;
    n9 = __ROR4__(0xB11924E1 * n8, 5);
    n10 = n9 + 0x27100001;
    n6 = n10;
    if ( str._Myres < 16 )
      v10 = &str;
    v10->_Bx._Buf[i++] = n10 % 25 + 'a';
  }
  while ( i < n7 );
}
v13 = (std::basic_string *)str._Bx._Ptr;
if ( str._Myres < 0x10 )
  v13 = &str;
v13->_Bx._Buf[i] = '.';
v14 = (std::basic_string *)str._Bx._Ptr;
n11 = __ROR4__(0xB11924E1 * n6, 5);
n12 = (n11 + 0x27100001) % 14u;
if ( str._Myres < 0x10 )
  v14 = &str;
v14->_Bx._Buf[i + 1] = aRupweuinytpmusfrdeit[2 * n12];
v17 = (std::basic_string *)str._Bx._Ptr;
if ( str._Myres < 0x10 )
  v17 = &str;
v17->_Bx._Buf[i + 2] = aRupweuinytpmusfrdeit[2 * n12 + 1];
```

Pseudo-code of Locky C&C domain generation algorithm

Communication with a C&C is performed using the HTTP protocol. The Trojan sends a POST request to an address with the format http://<cnc_url>/main.php; the transmitted data is encrypted with a simple symmetric algorithm.

Let’s have a look at the possible types of transmitted parameters.

1. 1

   Notification about infection and request for key.
   id=<infection id>
   &act=getkey&affid=<partner id contained in the Trojan’s body>
   &lang=<language of the operating system>
   &corp=<whether the OS is a corporate OS>
   &serv=<whether the OS is a server OS>
   &os=<OS version>

&sp=<version of OS service pack>
&x64=<whether the OS is 32- or 64-bit>

Judging by the affid parameter, Locky is distributed via an affiliate, or partnership, program.

2. 2

Sending list of encrypted paths.
id=<infection id>
&act=report&data=<list of paths>

For each disk drive it has handled, the Trojan sends the C&C a list of all paths to all encrypted files.

3. 3

Sending statistics for each handled disk drive.
id=<infection id>
&act=stats&path=<path>
&encrypted=<number of files encrypted>
&failed=<number of errors>
&length=<total size of encrypted files>

It should be noted that the cybercriminal collects very detailed statistics for each infection. Other ransomware families that we analyzed earlier were not this thorough at collecting statistics.

Countermeasures

Kaspersky Lab products protect against the Locky ransomware Trojan at all stages of the attack:

- The anti-spam module detects emails sent by the Trojan's distributors;
- Script loaders are detected by static and heuristic signatures of email and file antivirus with the verdicts Trojan-Downloader.MSWord.Agent, Trojan-Downloader.JS.Agent, HEUR:Trojan-Downloader.Script.Generic;
- The Trojan's executable file is detected by file antivirus signatures as Trojan-Ransom.Win32.Locky;
- Unknown samples of Locky are proactively detected by the System Watcher module with the verdict PDM:Trojan.Win32.Generic.

Preventing infections

Locky is a typical ransomware Trojan, and it exhibits no major differences from other ransomware families in its internal arrangement or its principles of operation. However, it caught the attention of researchers because it was so active and so widespread. According to KSN data, Kaspersky Lab products have blocked Locky attacks in over 100 countries around the world – no other ransomware Trojan to date has attacked so many countries at once.

To protect yourself from this ransomware Trojan, follow these preventive measures:

- Do not open attachments in emails from senders you don't know;

- Back up your files on a regular basis and store the backup copies on removable storage media or in cloud storages – not on your computer;
- Regularly run updates for your antivirus databases, operating system and other software installed on your computer;
- Create a separate network folder for each user when managing access to shared network folders.

For more detailed information about protection from ransomware Trojans, please follow this link.

Source: https://securelist.com/locky-the-encryptor-taking-the-world-by-storm/74398/