Offline Ransomware Encrypts Your Data without C&C Comms

By bferrite

Published: 2015-11-04 · Archived: 2026-05-05 02:11:29 UTC

Early in September, Check Point obtained a sample of a ransomware. When the sample was run, the following message, written in Russian, appeared:

Translation:

“Your files are encrypted, if you wish to retrieve them, send 1 encrypted file to the following mail address:

Seven_Legion2@aol.com

ATTENTION!!! You have 1 week to mail me,
after which the decryption will become impossible!!!!”

All personal files were indeed encrypted, with each file renamed to the following format:

email-[address to contact].ver-[Ransomware internal version].id-[Machine identifier]-[Date & Time][Random digits].randomname-[Random name given to the encrypted file].cbf

Example:

email-Seven_Legion2@aol.com.ver-CL 1.0.0.0.id-NPEULAODSHUJYMAPESHVKYNBQETHWKZOBQFT-10@6@2015 9@53@19 AM5109895.randomname-EFWMERGVKYNBPETHVKZNBQETHWKZNB.RGV.cbf

When running, the ransomware does not interact with the user, other than changing the wallpaper. Furthermore, while most known ransomware requires Internet connection and successful communication to their C&C servers before initiating the encryption, this sample does not need Internet connection to encrypt files and display the ransom message. This means that there is no key exchange between the infected machine and the attacker, which eliminates one option of stopping the attack.

Check Point reached out anonymously to the attacker’s email, and received a reply requesting a payment of 20,000 Russian Ruble (approx. $300) on the same day or 25,000 (approx. $380) on the following day, to receive a decryption program and key.

As the behavior is quite different from most known samples, we decided to explore the ransomware in greater depth, in both the intelligence and technical aspects.

Intelligence Analysis

During our research, we found many online references to this ransomware, especially in Russian language forums. This ransomware family appears to have been around for over a year, with the first reference in June 2014 to the

original version (no associated version number). Since then, 11 new versions have been reported. The chart below shows the version history (dates according to first appearance in forums or reports online):

Note: Between March and April 2015, the file version schema changed. A “CL” prefix was added and the version numbers were restarted. A parallel change was also made to the format of the encrypted file name. The older format (pre-April 2015) is as follows:

[original file name].id-{[Machine identifier]-[Date]-[Time][Random digits]}-email-[address to contact]-ver-[Ransomware internal version].cbf

Example:

excel4.xls.id-{AHMSYEKQVBHMSYEJPVAGMSXDJOUAGLRWCHNT-11@09@2014 00@47@473042530}-email-ivanivanov34@aol.com-ver-4.0.0.0.cbf

*For the original version, which did not have a specific version number, the format was identical to the older format, except for the lack of the version section at the end of the file name. For example:

001.jpg.id-{NKYVGVEZRTCYMVENMHRALUDMXGPKJSBXWFAJ-03.09.2014 1@45@403928355}-email-sishelp100@gmail.com.cbf

Differences between the two formats:

- The older encrypted file name includes the name of the original file. In the current format, the file is given a random name.
- The older format has the email address and the version at the end. The current format has them at the beginning.

Many email addresses, mainly AOL and Gmail accounts (but also others) have been associated with this ransomware:

Note the email address madeled@mail.ru, which is the only one found to be associated with a Russian email provider, and was also one of the email addresses associated with the original version of this ransomware. It has not appeared after version 4.0.0.0.

This ransomware (at least specific versions of it) has been given unique names by different vendors:

- **Ransomcrypt.U** (Symantec) – See https://www.symantec.com/security_response/writeup.jsp?docid=2015-092211-0927-99&tabid=2

This includes a comprehensive analysis of the ransomware behavior on the machine (not including the encryption mechanism).

- **Win32.VBKryjetor.wfa** (Kaspersky) – Refers to version CL 1.0.0.0 (for the specific sample we analyzed)
- **Ninja Ransomware** (Enigma Software) – Refers to the email gaiver@aol.com, although there is no apparent difference from other emails.
- **Troj/Agent-AOTR** (Sophos) – Refers to version CL 1.0.0.0

- **Troj/Drop-HQ** (Sophos) – Refers to version CL 1.0.0.0
- **Troj/Ransom-AZT** (Sophos) – Refers to version CL 1.0.0.0
- **Troj/Ransom-BGX** (Sophos) – Refers to version CL 1.0.0.0
- **Troj/Ransom-BJQ** (Sophos) – Refers to version CL 1.0.0.0
- **Troj/Ransom-BJV** (Sophos) – Refers to version CL 1.0.0.0
- **Troj/Agent-ANBL** (Sophos) – Refers to version CL 0.0.1.0
- **Troj/Ruftar-H** (Sophos) – Refers to version CL 1.0.0.0
- **Troj/VB-IHK** (Sophos) – Refers to version 6.1.0.0.b
- **Mal/Delp-AI** (Sophos) – Refers to version 4.0.0.0

Technical Analysis (Encryption)

Ransomware Details

The ransomware sample investigated by Check Point was from version CL 1.0.0.0 (as can be seen in the encrypted file name). It uses a protector that was written in *Visual Basic* compiled language. To unpack the payload, the ransomware restarts its own process using section mapping and overwrites four times. The payload that is responsible for file encryption is most likely written in *Delphi* language using some additional *Pascal* modules (for example, *FGInt* that is used to represent large numbers). We mention this fact as it is not usual for ransomware to utilize *Pascal*-based languages. The ransomware does not contain much functionality except for the file encryption capability.

Encryption Functionality

The encryption functionality is built with several layers of encoding and encryption, including two separate levels of RSA:

1. The beginning (first 30000 bytes) of each file is encrypted using two buffers of digits and letters that are randomly generated on the infected machine. The encryption process includes taking each original byte along with one byte from each of the randomly generated buffers and performing mathematical operations on them.
2. The remainder of each file (if it exists) is encrypted using an RSA public key (“local”) that is randomly generated on the infected machine, along with the matching local RSA private key required for decryption of the data.
3. The randomly generated buffers and the local RSA private key that are required for decryption are added as metadata to each encrypted file, and are then encrypted using three hardcoded RSA 768 public keys that the offender created in advance (“remote”). The matching remote RSA private keys required to unlock the metadata are located on the attacker’s side.

Due to this functionality, the ransomware is able to encrypt all files locally without connecting to a C&C server. Once the attacker receives a file from the infected machine, he can easily decrypt the metadata using his remote RSA private keys, and find the buffers and local RSA private key that were randomly generated on the infected machine which can be used to decrypt the file.

It is not feasible to try to decrypt the remote RSA encryption without the remote private key. The necessary time frame would be approximately 2 years and would involve using many computers. Therefore, paying the ransom to get the decryption application and the decryption keys from the attacker seems to be the only way to recover the encrypted files.

For More Details Read: Check Point Technical Report
