Nokoyawa Ransomware | New Karma/Nemty Variant Wears Thin Disguise

sentinelone.com/labs/nokoyawa-ransomware-new-karma-nemty-variant-wears-thin-disguise/

Antonis Terefos

Executive Summary

- At the beginning of February 2022, SentinelLabs observed two samples of a new Nemty variant dubbed “Nokoyawa” (sample 1, 2).
- SentinelLabs consider Nokoyawa to be an evolution of the previous Nemty strain, Karma.
- The developers have attempted to enhance code responsible for excluding folders from encryption, but SentinelLabs analysis finds that the algorithm contains logical flaws.
- In March, TrendMicro suggested this ransomware bore some relation to Hive. We assess that Hive and Nokoyawa are different and that the latter is not a rebrand of Hive RaaS.

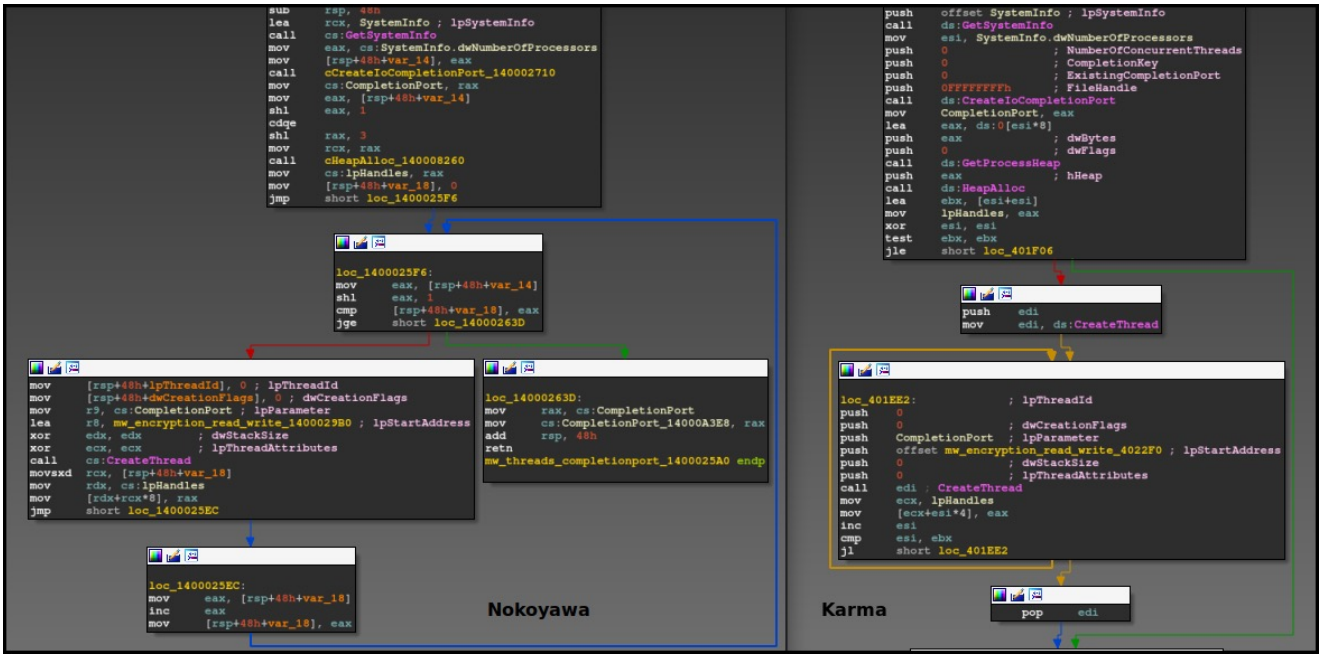
Overview

In this post, we take a broader look at the similarities between Nokoyawa and Karma ransomware. Previous researchers have highlighted similarities in the attack chain between Nokoyawa and Hive ransomware, concluding that “Nokoyawa is likely connected with Hive, as the two families share some striking similarities in their attack chain, from the tools used

to the order in which they execute various steps." Our analysis contradicts that finding, and we assess Nokoyawa is clearly an evolution of Karma (Nemty), bearing no major code similarities to Hive.

Nokoyawa and Karma Variant Similarities

Both Nokoyawa and Karma variants manage multi-threaded encryption by creating an input/output (I/O) completion port (CreateIoCompletionPort), which allows communication between the thread responsible for the enumeration of files and the sub-threads ("2 * NumberOfProcessors") responsible for the file encryption.



Nokyoawa (left) vs Karma, initialization of encryption threads

In both cases, public keys for the encryption and ransom note are encoded with Base64.

Like Karma, Nokoyawa accepts different command line parameters, although in the latter they are documented by the developer via a `-help` command.

```
C:\Users\                    ml.exe -help

NOKOYAWA.exe (Encrypt all local, network drives without network shares)
NOKOYAWA.exe -network (Encrypt all local, network drives including network shares)
NOKOYAWA.exe -file filePath (Encrypt only selected file)
NOKOYAWA.exe -dir dirPath (Encrypt selected directory and sub-directories)
```

Nokoyawa command line support

Aside from the `-help` command, three other commands (`-network` , `-file` , and `-dir`) are also provided.

Parameter	Functionality
-help	Prints command line options for execution of ransomware.

-network	Encrypts local and network shares.
-file	Encrypts specified file.
-dir	Encrypts specified directory.

If the ransomware is executed without any parameter, it then encrypts the machine without enumerating and encrypting network resources.

One new parameter not observed in the Karma version is `-network`, which is responsible for encrypting network shares. Network enumeration is achieved by calling WNetOpenEnumW, WNetEnumResourceW, and WNetCloseEnum.

There are no significant similarities between the ransom notes except the use of email for contact points. Karma variants contained an `.onion` link that was also present in the Karma ransom note. We did not find any `.onion` links in Nokoyawa code or ransom note.

The Nokyoawa ransom note:

```
Dear usernamme, your files were encrypted, some are compromised.
Be sure, you can't restore it without our help.
You need a private key that only we have.
Contact us to reach an agreement or we will leak your black shit to media:

[email protected]
[email protected]

亲爱的用户名，您的文件已加密，有些已被泄露。
请确保，如果没有我们的帮助，您将无法恢复它。
您需要一个只有我们拥有的私钥。
联系我们以达成协议，否则我们会将您的黑屎泄露给媒体：

[email protected]
[email protected]
```

The Karma ransom note:

```
Your network has been breached by Karma ransomware group.
We have extracted valuable or sensitive data from your network and encrypted the data
on your systems.
Decryption is only possible with a private key that only we posses.
Our group's only aim is to financially benefit from our brief acquaintance,this is a
guarantee that we will do what we promise.
Scamming is just bad for business in this line of work.
Contact us to negotiate the terms of reversing the damage we have done and deleting
the data we have downloaded.
We advise you not to use any data recovery tools without leaving copies of the
initial encrypted file.
You are risking irreversibly damaging the file by doing this.

If we are not contacted or if we do not reach an agreement we will leak your data to
journalists and publish it on our website.

http://3nvzqyo6l4wkrzumzu5aod7zbosq4ipgf7ifgj3hsvbcr5vcasordvqd.onion/

If a ransom is payed we will provide the decryption key and proof that we deleted you
data.
When you contact us we will provide you proof that we can decrypt your files and that
we have downloaded your data.

How to contact us:
[email protected]
[email protected]
[email protected]
```

The ransom note filename uses a similar format as the previous versions: `<ransom_extension>_<note_name>.txt`.

	Nokoyawa	Karma
ransom_extension	“NOKOYAWA”	“KARMA” & “KARMA_V2”
note_name	“_readme.txt”	“-ENCRYPTED.txt”

The `<ransom_extension>` string has been used for many different functions, including:

- file extension of encrypted files
- appended as data to an encrypted file
- ransom note filename part
- mutex (the NOKOYAWA variant is observed to not make use of Mutexes)
- to denote files to be excluded from further processing (e.g., to avoid running in a loop)

Nokoyawa’s Flawed Encryption Routine

During the file and folder enumeration, the new variant creates a hash of the enumerated folder and compares it to those of excluded folders. However, this custom hashing algorithm appears to have flaws as it doesn't seem logical nor does it appear to work as expected.

Below is a Python representation of the hashing algorithm.

```
def nokoyawa_dir_hashing(folder):
    folder_len = len(folder)
    # to unicode
    folder = '\x00'.join([c for c in folder])
    # initial hash value
    nhash = 0x1505
    i = 0
    while i < folder_len:
        c = ord(folder[i])
        # to capital
        c = c if c < 0x61 else c - 0x20
        # hashing
        nhash = ((nhash << 5) + nhash) + c
        # flawed logic, taking only null bytes after 1st character.
        i += 2 if not c else 1
    return nhash & 0xFFFFFFFF
```

The implementation of this flawed hashing algorithm in some cases results in excluding multiple folders. Logically, one would expect there to be a 1:1 correlation between a hash and the folder to be excluded. However, the flawed code effectively makes it possible for multiple folders to be excluded based on a single hash. This code does not appear in Karma variants, which instead contain hardcoded strings denoting which folders to ignore during encryption.

The following table shows which folders the developers intended to skip during encryption.

Hash	Folders Intended To Be Excluded
0x11f299b5	program files
0x78fb3995	program files (x86)
0x7c80b426	appdata
0x7c8cc47c	windows
0xc27bb715	programdata
0xd6f02889	$recycle.bin

For extensions, the ransomware doesn't have any hashing algorithm and compares the raw strings with the extracted extension of the file. The excluded extensions are `.exe`, `.dll`, and `.lnk`. Files containing "NOKOYAWA" are also excluded.

Both Nokoyawa and Karma variants dynamically load `bcrypt.dll` and call BCryptGenRandom in order to generate 0x20 random bytes. They generate an ephemeral Sect233r1 key pair using the generated random bytes as the seed. The malware then uses the private ephemeral key and the public embedded key to generate a shared Salsa20 key, which is subsequently used for the file encryption. The Salsa20 nonce is hardcoded as `“lvcelvce”` in Nokoyawa, whereas in the Karma version it was `"11111111"`.

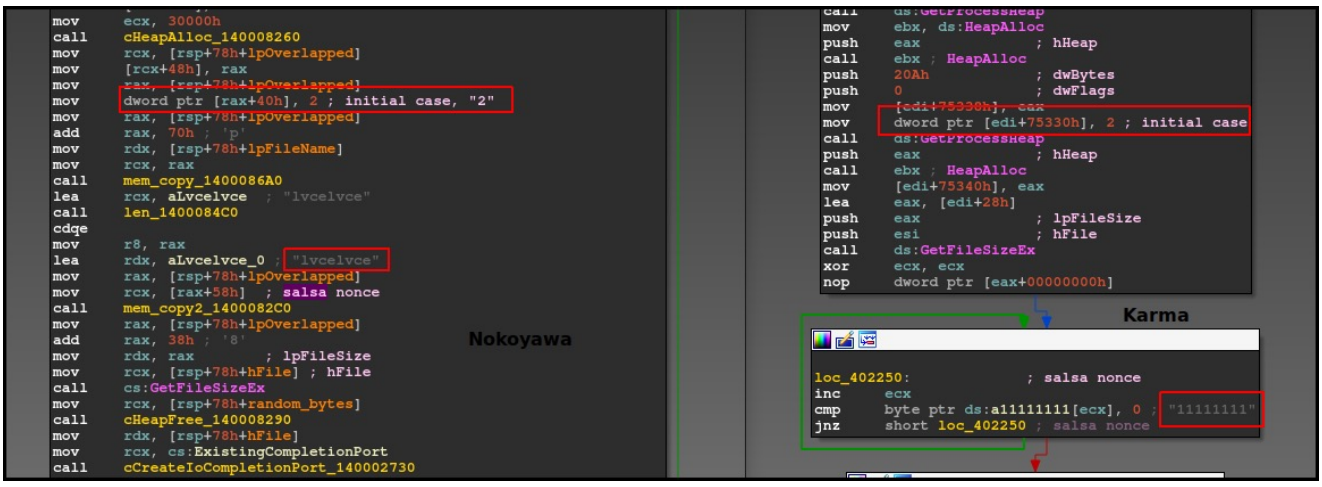
An I/O completion packet is sent to the thread responsible for encryption. The packet includes the following:

- File handle
- File size
- File data
- Salsa20 key
- Salsa nonce
- public ephemeral key

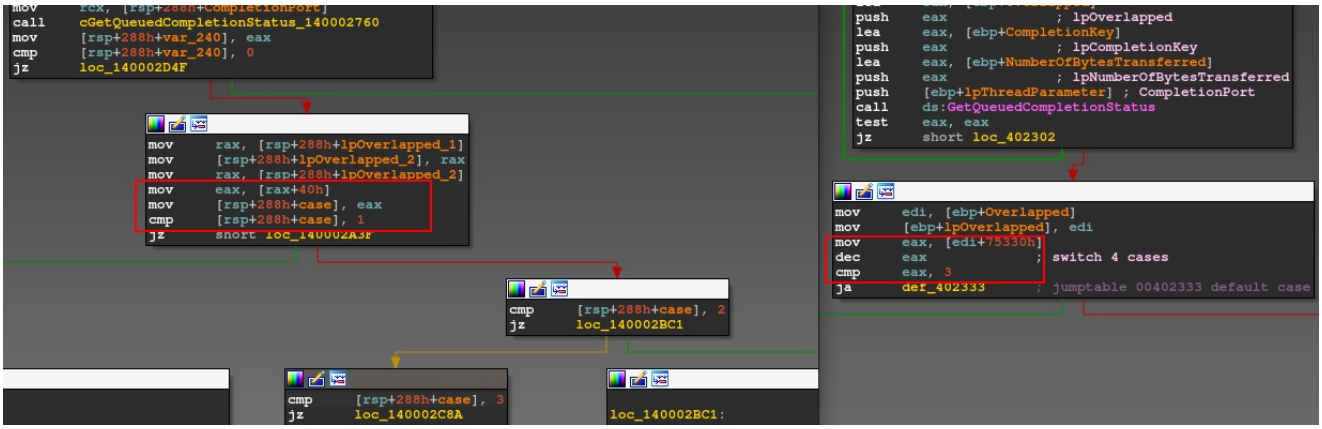
The encryption thread has a switch containing four cases, as follows:

- **Case 1:** Writes encrypted content and decryption struct to file and appends "extension"/"variant name".
- **Case 2:** Calculates validation SHA1 hash and encrypts file data with Salsa20.
- **Case 3:** Closes File Handle and moves files with the new extension.
- **Case 4:** Exits.

In both variants, the initial switch case is "2".



Initial case, encryption thread



Encryption thread case handler

During Case 2, the malware adds a SHA1 checksum, which is possibly validated during the decryption phase. The method runs through the following logic:

- Allocates 0x13 bytes (0x14 required for SHA1)
- XORs Salsa key with a buffer of "6".
- Concatenates file data to XORed Salsa key
- Calculates SHA1.
- XORs Salsa key with a buffer of "\".
- Concatenates SHA1 hash to the second XORed Salsa key.
- Calculates validation SHA1.
- Validation SHA1 hash first 0x13 bytes are added to the encrypted file struct

Files encrypted by Nokoyawa will have the following structure.

```
struct nokoyawa_encrypted_file
{
    BYTE  encrypted_file_data[file_size],  // using salsa20
    BYTE  public_ephemeral_key[0x40],  // Sect233r1
    BYTE  validation_hash[0x13],  // last byte is chopped
    STRING ransomware_extension
}
```

The private key required for decryption is held by the attacker. When made available to the victim, the decryption routine reads the struct, extracts the public ephemeral key and generates the Salsa 20 key using the private key. The encrypted data is then decrypted with the key and validated by replicating the validated hash.

Conclusion

Nokoyawa code similarity and structure suggest it to be an evolution of the previous Nemty strain, Karma. This appears to be another attempt from the developer to confuse attribution. At this time, the actor appears not to have or provide any onion leak page.

SentinelLabs could not validate previous research suggesting Nokoyawa is related to Hive. Given the lack of code similarities between the two and the lack of further correlating data, we can only suggest that earlier researchers' findings may be explained by the possibility of an affiliate using both Hive and Nokoyawa.

SentinelLabs continues to follow and analyze the development of Nemty ransomware variants.

Indicators of Compromise

Karma Ransomware SHA1

960fae8b8451399eb80dd7babcc449c0229ee395

Nokoyawa Ransomware SHA1

2904358f825b6eb6b750e13de43da9852c9a9d91
2d92468b5982fbbb39776030fab6ac35c4a9b889
32c2ecf9703aec725034ab4a8a4c7b2944c1f0b7

Nokoyawa Ransom Note Email Addresses

Nokoyawa YARA Rule

```
rule Nokoyawa_Nemty
{
    meta:
        author = "@Tera0017"
        description = "Nokoyawa, Nemty/Karma ransomware variant"
        Reference = "https://www.sentinelone.com/labs/nokoyawa-ransomware-new-karma-
nemty-variant-wears-thin-disguise/"

    strings:
        $code1 = {B8 (41| 43) 00 00 00 [10-30] 83 F8 5A}
        $code2 = {48 8B 4C 24 08 F0 0F C1 01 03 44 24 10}
        $code3 = {83 E8 20 88 [7] 48 C1 E0 05 48 03 44 24}
        $code4 = {48 C7 44 24 ?? 05 15 00 00}
        $string1 =
"RGVhciB1c2VybmFtbWUsIHlvdXIgZmlsZXMgd2VyZSBlbmNyeXB0ZWQsIHNvbWUgY" ascii
        $string2 = "-network" fullword wide
        $string3 = "-help" fullword wide
        $winapi1 = "PostQueuedCompletionStatus" fullword ascii
        $winapi2 = "GetSystemInfo" fullword ascii
        $winapi3 = "WNetEnumResourceW" fullword ascii
        $winapi4 = "GetCommandLineW" fullword ascii
        $winapi5 = "BCryptGenRandom" fullword ascii

    condition:
        all of ($winapi*) and 4 of ($code*, $string*)
}
```