

Nokoyawa Ransomware: Rust or Bust | Zscaler

By Brett Stone-Gross

Published: 2022-12-20 · Archived: 2026-04-05 20:57:47 UTC

Key Points

- *Nokoyawa* is a 64-bit Windows-based ransomware family that emerged in February 2022
- The threat group behind *Nokoyawa* performs double extortion ransomware attacks: exfiltrating sensitive information from organizations, followed by file encryption and a ransom payment demand
- *Nokoyawa* was initially written in the C programming language using Elliptic Curve Cryptography (ECC) with SECT233R1 and Salsa20 for file encryption
- In September 2022, *Nokoyawa* was rewritten in the Rust programming language using ECC with the Curve25519 and Salsa20 for file encryption
- The Rust-based *Nokoyawa* ransomware 2.0 provides threat actors with runtime flexibility via a configuration parameter that is passed via the command-line

Nokoyawa ransomware was discovered in February 2022, sharing code with another ransomware family known as [Karma](#). *Nokoyawa* ransomware's [lineage](#) can further be traced back to Nemty ransomware. The original version of *Nokoyawa* ransomware was written in the C programming language and file encryption utilized asymmetric Elliptic Curve Cryptography (ECC) with Curve SECT233R1 (*a.k.a.* NIST B-233) using the [Tiny-ECDH](#) open source library combined with a per file Salsa20 symmetric key. *Nokoyawa* ransomware 2.0 still uses Salsa20 for symmetric encryption, but the elliptic curve was replaced with Curve25519.

Nokoyawa 2.0 was developed using the Rust programming language and appears to have been created in late September 2022. *Nokoyawa* is not the first ransomware family to be rewritten in Rust. Previously, the developers of the ransomware families [Hive](#) and [Agenda/Oilin](#) ported their code from the Go (*a.k.a.* Golang) programming language to Rust. In addition, the author of [RansomExx](#) converted the ransomware code from C++ to Rust. Another ransomware family compiled in Rust is *BlackCat/ALPHV*. The increase in the popularity of the Rust programming language may be due to its emphasis on performance and concurrency, which can make a ransomware's file encryption more efficient. Similar to the previous version of *Nokoyawa*, the Rust variant is compiled only for 64-bit versions of Windows.

This blog provides a technical analysis of *Nokoyawa* 2.0 including its new configuration, encryption algorithms, and data leak site.

Technical Analysis

Nokoyawa 2.0 cannot be executed without providing the required command-line arguments. When run without arguments, *Nokoyawa* will print the following help message shown in Figure 1.

```
How to run:
nokoyawa.exe --config <base64 encoded config> (to start full encryption)
nokoyawa.exe --config <base64 encoded config> --file <filePath> (encrypt selected file)
nokoyawa.exe --config <base64 encoded config> --dir <dirPath> (encrypt selected directory)
```

Figure 1. Nokoyawa 2.0 ransomware command-line help

The command-line arguments *--file* (to encrypt a single file) and *--dir* (to encrypt a directory) are identical to the previous version of Nokoyawa. However, Nokoyawa 2.0 requires a configuration file to execute the ransomware via the *--config* command-line argument. The configuration parameter is a Base64 encoded JSON object that has the following keys and values shown in Table 1.

Key	Value Format	Description
<i>NOTE_NAME</i>	(will be appended with .txt)	Ransom note filename
<i>NOTE_CONTENT</i>	Base64 encoded text	Ransom note content
<i>EXTENSION</i>	(without a period)	Encrypted file extension (also used as the Salsa20 nonce)
<i>ECC_PUBLIC</i>	Base64 encoded binary data	Curve25519 public key
<i>SKIP_EXTS</i>	JSON array	File extensions that will not be encrypted
<i>SKIP_DIRS</i>	JSON array	Directories that will not be encrypted

Table 1. Nokoyawa 2.0 configuration parameters

The decision by the Nokoyawa malware author to pass a full configuration file via the command-line is a unique design choice. This is indicative that the malware author has developed the ransomware to be flexible for multiple threat actors who are likely paid as affiliates to compromise organizations and deploy the ransomware in return for a percentage of the profit.

Encryption Algorithms

Nokoyawa 2.0 uses Curve25519 (via the open source [x25519_dalek](#) Rust library) for asymmetric encryption and Salsa20 for symmetric encryption. Nokoyawa first generates an ephemeral Curve25519 key pair. The ephemeral private key is used to generate a shared secret using a Diffie-Hellman key exchange with the Curve25519 public key that was passed via the *config* command-line parameter. The result is used as a Salsa20 key and the file extension is used as the nonce, which must be 8 bytes (as described in Table 1). Figure 2 shows an example file encrypted by Nokoyawa 2.0.

```

0007fef0 d3 a5 5f 72 c1 51 53 ed 3f 6f 6a 78 27 d2 a4 85 |.._r.QS.?ojx'...|
0007ff00 1b 63 ca 99 85 81 9c 9e 22 bd ae ee 8d cd b5 bd |.c.....".....|
0007ff10 f9 1a 9b 62 db 39 26 fd 05 53 77 bb b4 aa 77 44 |...b.9&..Sw...wD|
0007ff20 3e 35 1a 44 97 00 03 b9 01 e6 f4 9b 0b b5 bb 9d |>5.D.....|
0007ff30 40 df 3b eb ec 8f 5a d8 ce 65 27 6a 21 3d a8 dd |@.;...Z..e'j!=..|
0007ff40 57 50 b4 d8 b8 b6 17 bf aa a1 80 9f 67 a9 09 6f |WP.....g..o|
0007ff50 77 cd 71 52 fe 89 22 c8 4f e0 40 a0 73 7d f9 be |w.qR.."O.@.s)..|
0007ff60 85 84 86 37 eb 3b 9f b2 53 7d ce 7c 1e 2a 47 53 |...7.;..S}.|.*GS|
0007ff70 ce 83 4e 32 59 1a f4 95 a2 35 15 0e d5 3b 9a 2b |..N2Y....5...;.+|
0007ff80 e4 98 ab 58 92 bf 34 e2 3e 81 8e e2 23 33 79 bf |...X..4.>...#3y.|
0007ff90 c3 aa 3d f5 e7 12 ca d9 47 e8 02 84 6f 49 31 67 |...=.....G...oIlg|
0007ffa0 b0 82 1b 49 28 34 9d 02 77 53 24 bc 4e 2d cf 7e |...I(4..wS$.N-~|
0007ffb0 c6 b7 0d 74 ec 85 4f 00 44 61 35 5a 1f 00 d0 99 |...t..O.Da5Z....|
0007ffc0 e3 d7 68 96 9b 9a 5f 01 b7 85 4d 56 8d 79 24 6d |..h..._...MV.y$M|
0007ffd0 5b b6 2a 7d 15 e6 45 15 72 2c b7 29 b6 81 84 c5 |[*}..E.r,..)....|
0007ffe0 56 c8 ee c2 b0 5c 81 95 98 72 47 52 4a 54 df a3 |V....\...rGRJT..|
0007fff0 0b cc c2 30 f8 4a 5d 7f cd f7 81 96 1a 55 b5 f3 |...0.J].....U..|
00080000 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 |AAAAAAAAAAAAAAAAAA|
*
001570e0 52 24 a7 5b 78 0f d6 f4 e8 b7 97 25 cd f8 02 1d |R$. [x.....%....|
001570f0 43 1e ae 1b 6f fc 57 9b 70 c3 1c 7d 18 95 84 0d |C...o.W.p..)....|
00157100 6e 6f 6b 6f 79 61 77 61 |nokoyawa|

```

Figure 2. Nokoyawa 2.0 encrypted file content and footer

As shown in Figure 2, the 32-byte ephemeral public key (blue) and the 8-byte nonce (red) are appended as a 40-byte footer at the end of the encrypted file. Similar to most ransomware families, Nokoyawa encrypts the file in chunks based on the file's size. If the file's size is less than or equal to 0x80000 (524,288) bytes, the full file will be encrypted. Otherwise the code implements an algorithm that determines the number of blocks and the block offsets to encrypt in the file. Each block will be encrypted in chunks of 0x80000 bytes (yellow) followed by blocks of unencrypted bytes (green) as highlighted in Figure 2. Since Nokoyawa only partially encrypts files larger than 0x80000 bytes, encryption is very fast.

ThreatLabz has developed a proof-of-concept tool to decrypt files encrypted by Nokoyawa 2.0 if the Curve25519 private key is accessible. This decryption tool is available in our GitHub tools repository [here](#).

Ransom Note

As previously mentioned in Table 1, the Nokyawa ransomware note filename and content is passed via the *configuration* command-line parameter. An example Nokoyawa ransom note is shown in Figure 3.

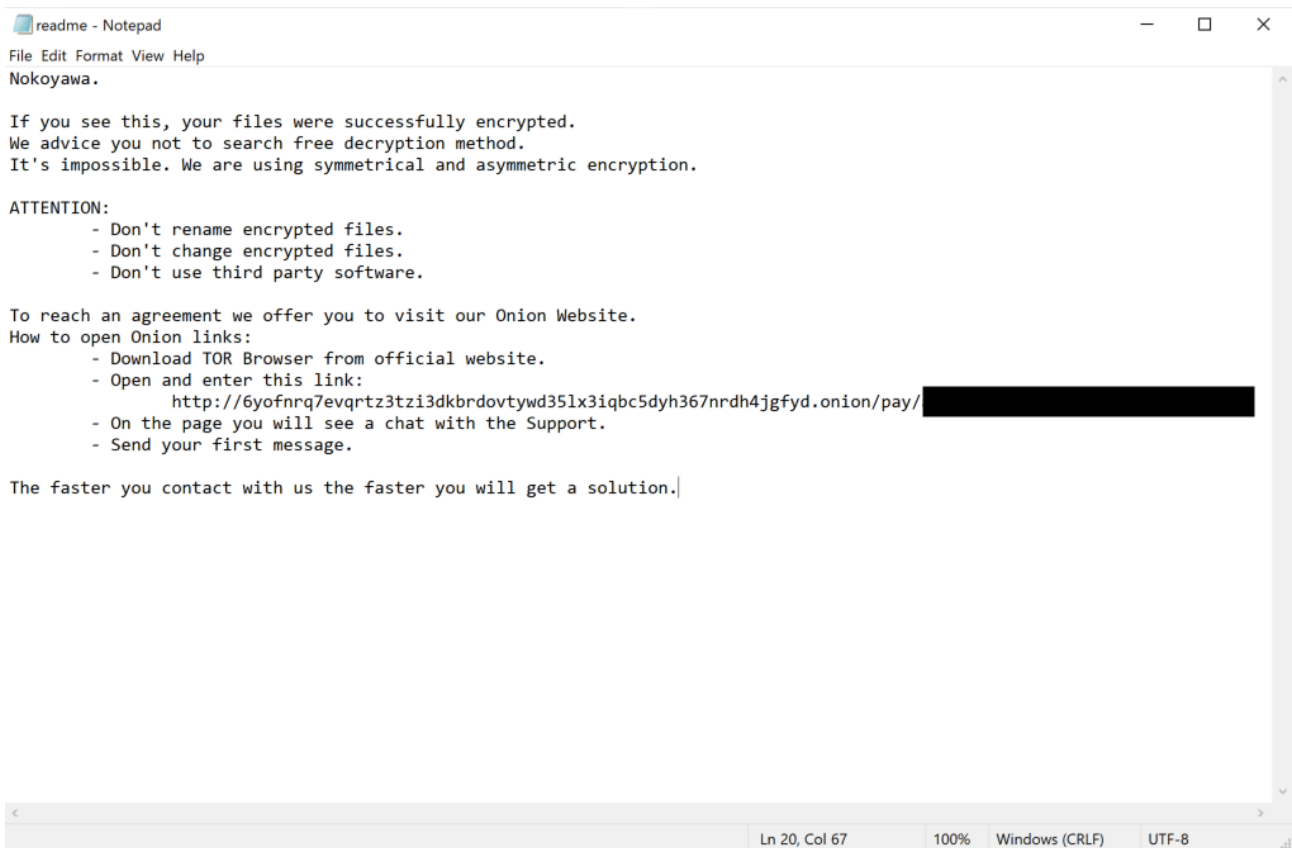


Figure 3. Nokoyawa ransom note

Ransom portal

Nokoyawa ransom notes contain a link to a TOR hidden service as shown in Figure 4.

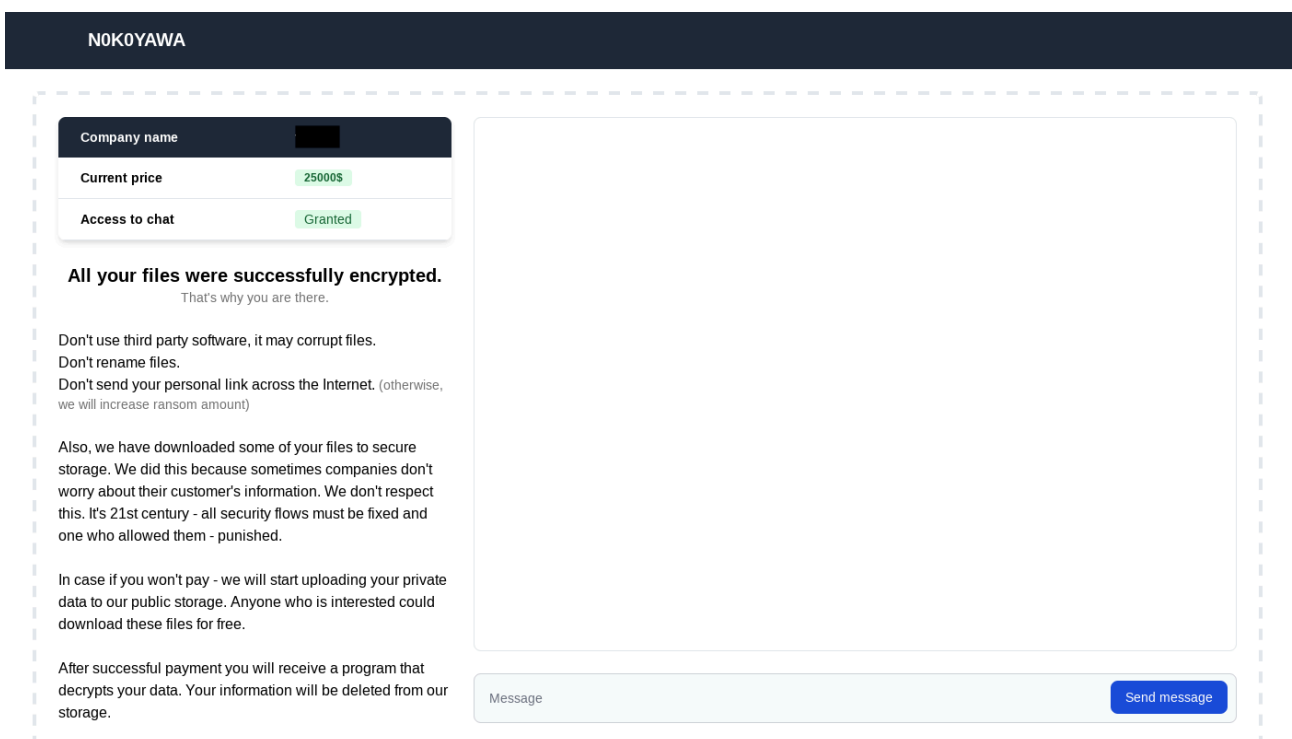


Figure 4. Nokoyawa ransom chat portal

The same TOR hidden service also hosts a data leak site. Currently, only one victim is listed on the site as shown in Figure 5. This may suggest that Nokoyawa is not currently compromising a large number of organizations, or the threat actors may only perform double extortion for a subset of victims.

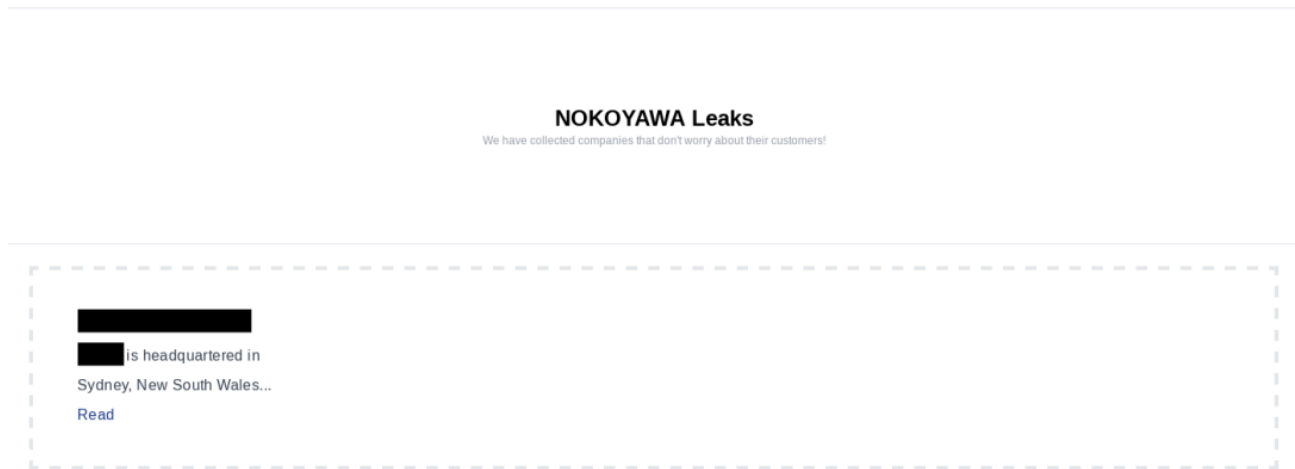
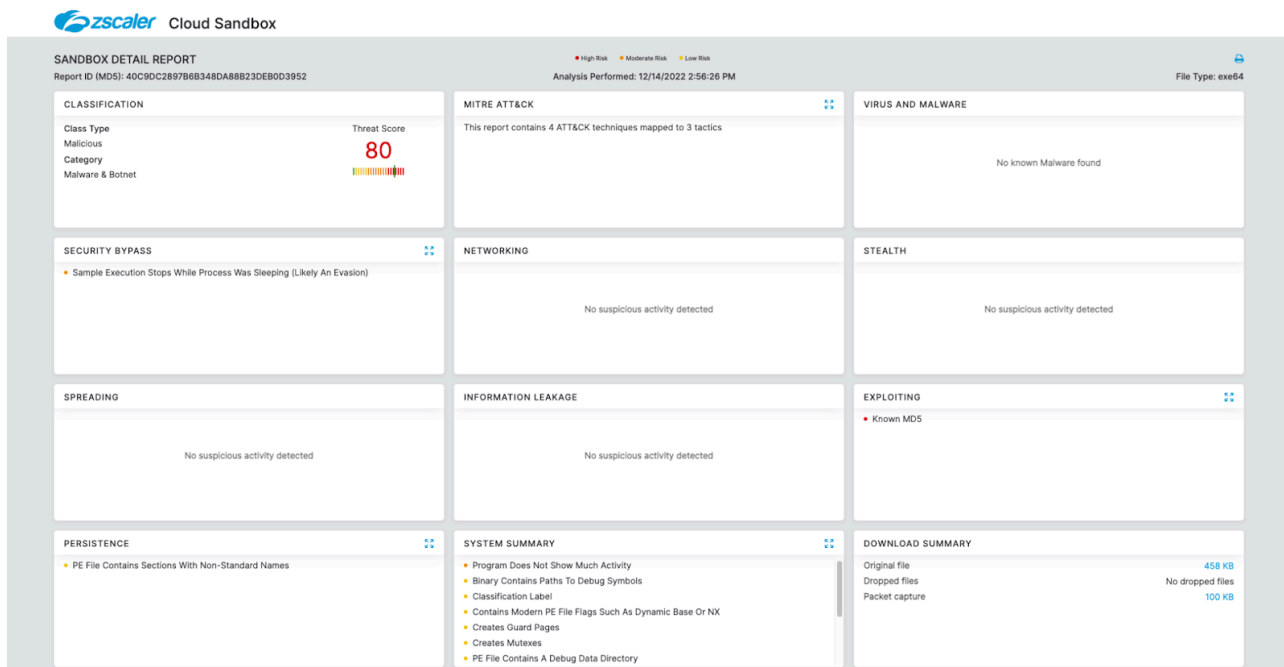


Figure 5. Nokoyawa leak site

Conclusion

The Nokoyawa threat actor continues to update the ransomware and launch new attacks. The development of Nokoyawa 2.0 using the Rust programming language is likely designed to improve file encryption speed and to better evade antivirus and EDR products. The group has long claimed to perform double extortion attacks without offering much proof, until now.

Cloud Sandbox Detection



In addition to sandbox detections, Zscaler’s multilayered cloud security platform detects indicators related to Nokoyawa at various levels with the following threat names:

- [Win64.Ransom.NOKOYAWA](#)

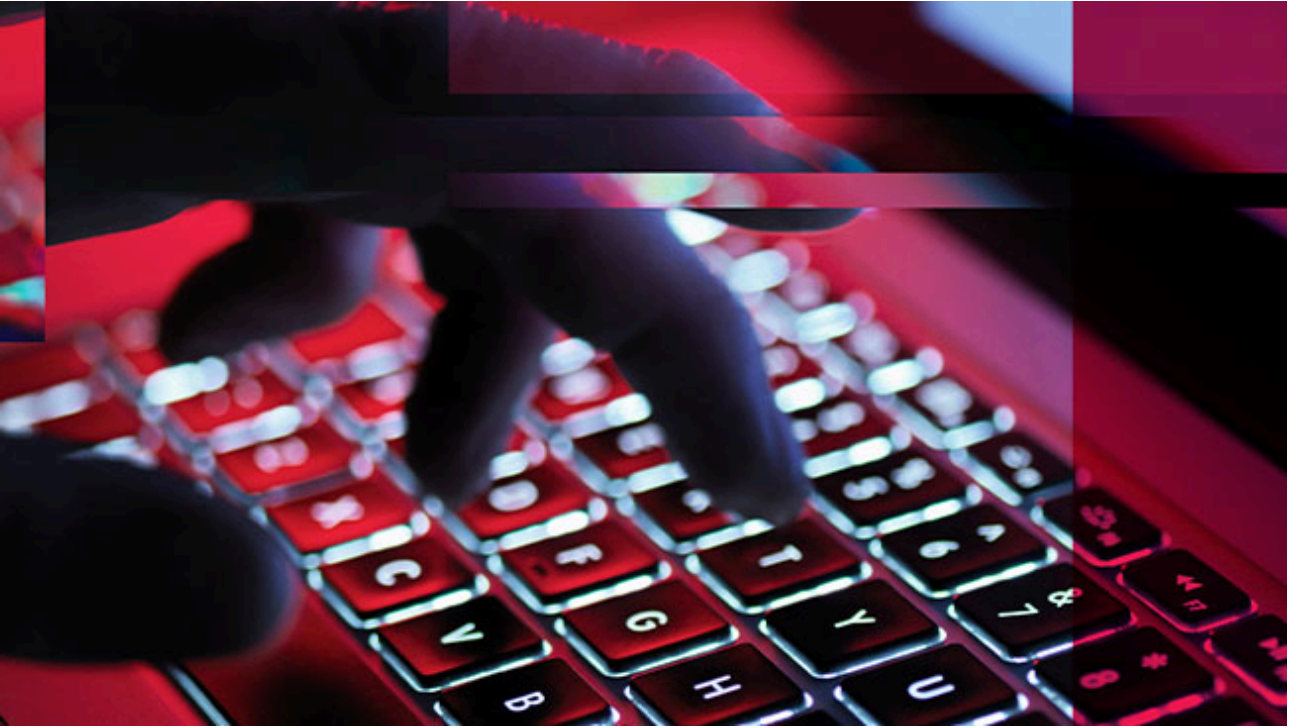
Indicators of Compromise

SHA256	Description
7095beafff5837070a89407c1bf3c6acf8221ed786e0697f6c578d4c3de0efd6	Nokoyawa ransomware Rust sample
47c00ac29bbaee921496ef957adaf5f8b031121ef0607937b003b6ab2a895a12	Nokoyawa ransomware Rust sample
259f9ec10642442667a40bf78f03af2fc6d653443cce7062636eb750331657c4	Nokoyawa ransomware Rust sample

Explore more Zscaler blogs

Zscaler ThreatLabz 2024 Phishing Report

The Threat Prevention Buyer's Guide



Source: <https://www.zscaler.com/blogs/security-research/nokoyawa-ransomware-rust-or-bust>