Nokoyawa Ransomware: Rust or Bust

zscaler.com/blogs/security-research/nokoyawa-ransomware-rust-or-bust

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 Nokoyama 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/Qilin* 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
NOTE_NAME	<filename> (will be appended with .txt)	Ransom note filename
NOTE_CONTENT	Base64 encoded text	Ransom note content
EXTENSION	<8 characters> (without a period)	Encrypted file extension (also used as the Salsa20 nonce)

ECC_PUBLIC	Base64 encoded binary data	Curve25519 public key
SKIP_EXTS	JSON array	File extensions that will not be encrypted
SKIP_DIRS	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 mulitiple 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.

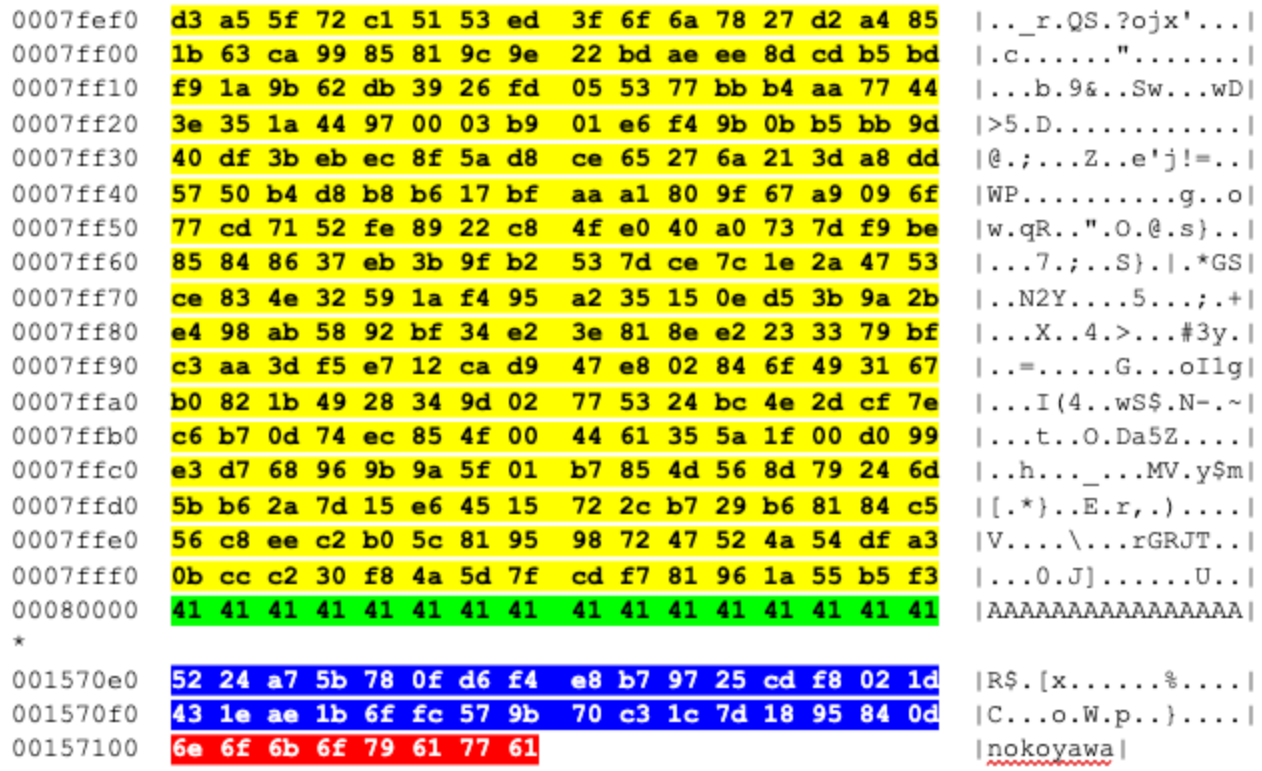


Figure 2. Nokayawa 2.0 encrypted file content and footer

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.

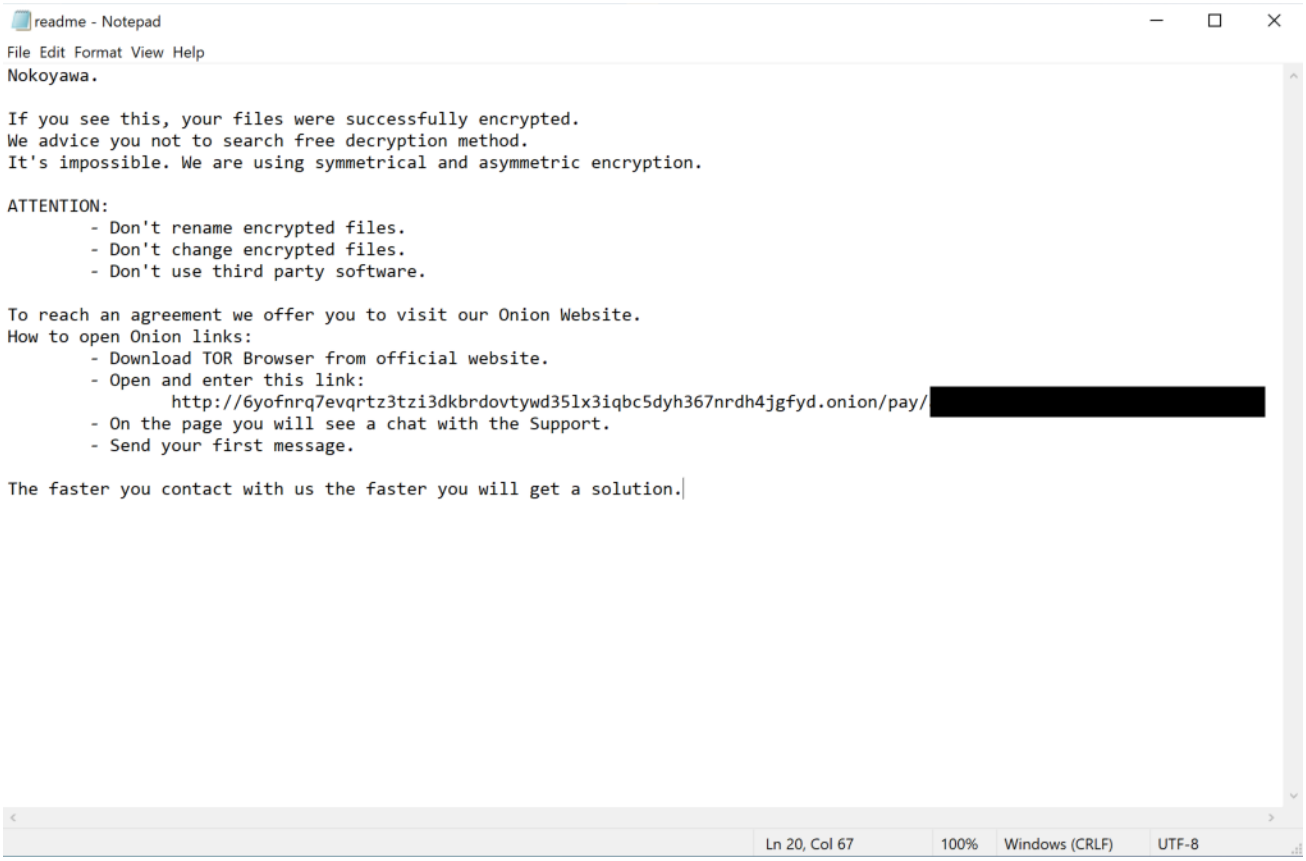


Figure 3. Nokoyawa ransom note

Ransom portal

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

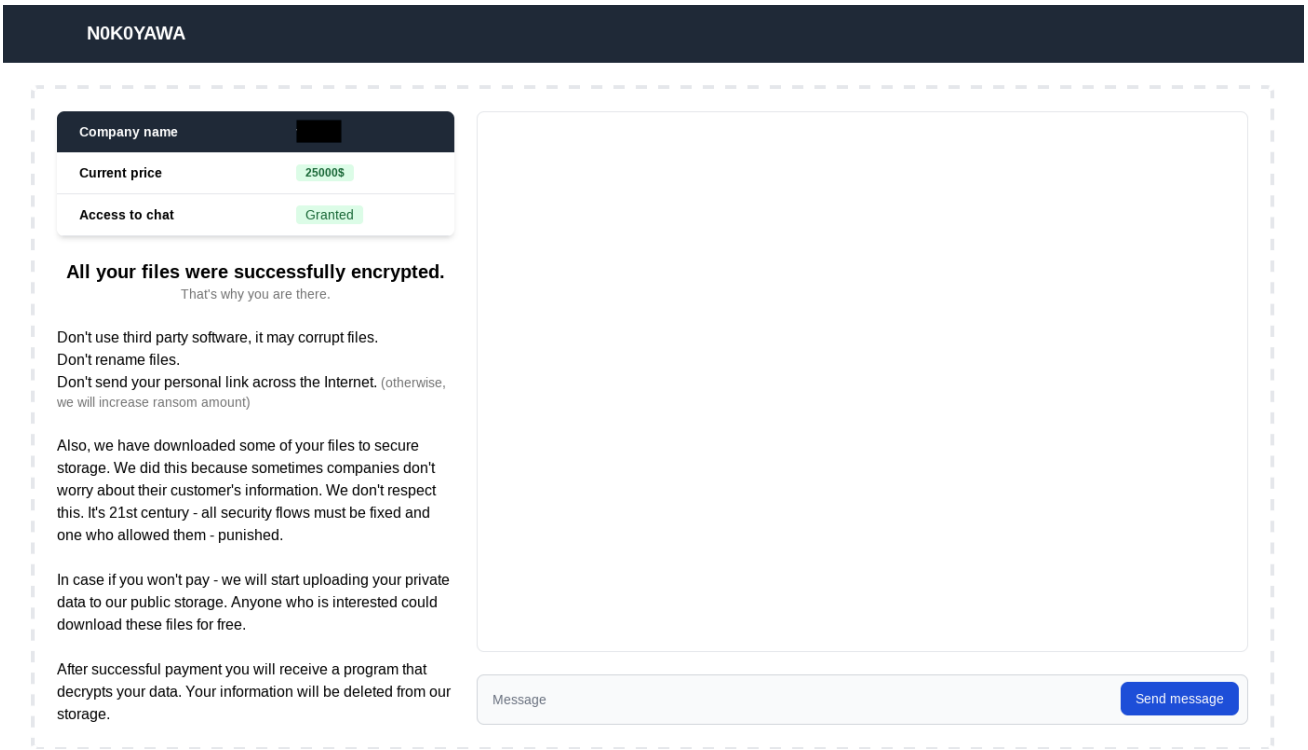


Figure 4. Nokoyawa ransom chat portal

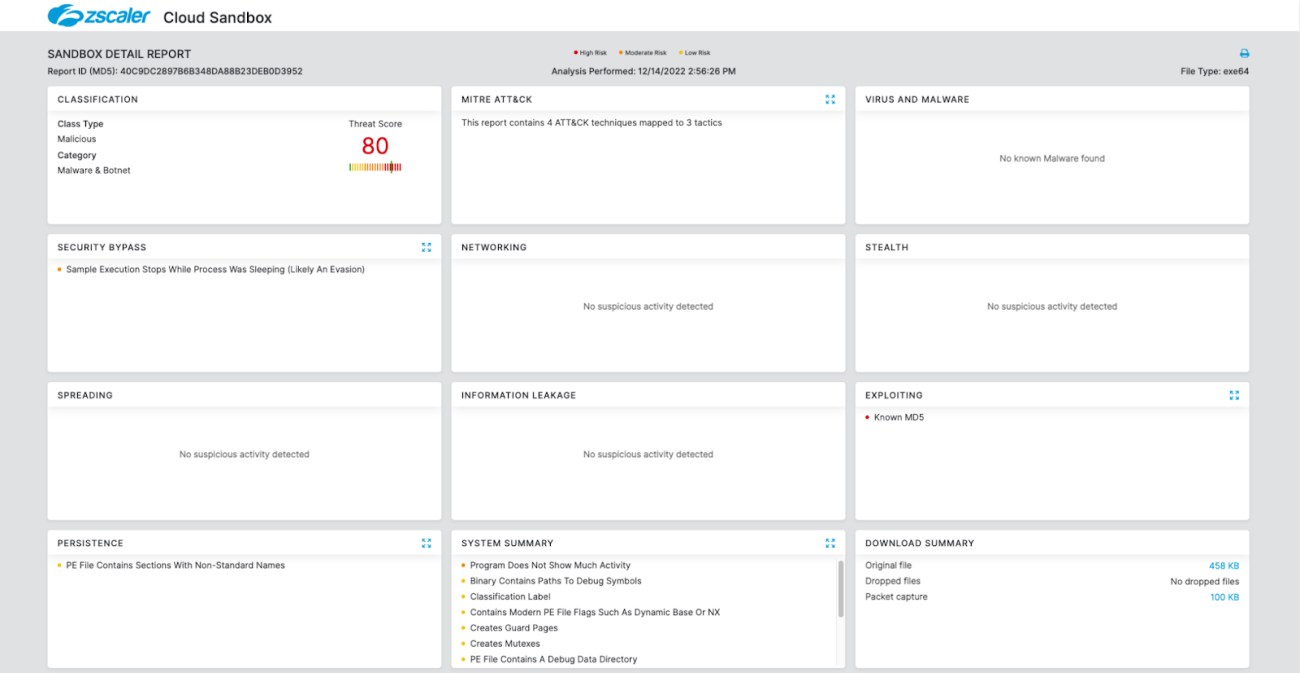
The same TOR hidden service also hosts a data leak site. Currently, only one victim is listed on the site as shown in Figure 5. This may suggest that Nokoyawa is not currently compromising a large number of organizations, or the threat actors may only perform double extortion for a subset of victims.

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

Security Research

Ransomware