Detect Nokoyawa ransomware With YARA Rule.

malgamy.github.io/malware-analysis/Nokoyawa/

December 21, 2022

4 minute read

On this page

How to write Yara rule for Nokoyawa ransomware

In the frist, we will work with 3 files that shared by zscaler

Introduction

Nokoyawa is a ransomware family that targets 64-bit Windows systems. It was first identified in February 2022 and is known for its use of double extortion tactics, which involve exfiltrating sensitive data from targeted organizations before encrypting files and demanding a ransom payment. The initial version of Nokoyawa was written in C programming language and used Elliptic Curve Cryptography (ECC) with SECT233R1 and Salsa20 for file encryption. In September 2022, a revised version of Nokoyawa was released, which was rewritten in Rust programming language and utilized ECC with Curve25519 and Salsa20 for file encryption. This new version, known as Nokoyama 2.0, includes a configuration parameter that can be passed via the command-line, providing threat actors with greater flexibility at runtime.

IOCs

- 7095beafff5837070a89407c1bf3c6acf8221ed786e0697f6c578d4c3de0efd6
- 47c00ac29bbaee921496ef957adaf5f8b031121ef0607937b003b6ab2a895a12
- 259f9ec10642442667a40bf78f03af2fc6d653443cce7062636eb750331657c4

Loading sample with IDA pro

I manually write Yara rules by using IDA Pro to load samples and examine their strings for unique characteristics relevant to a specific family. From this analysis, I can use the identified strings to craft effective Yara rules.

Strings with IDA

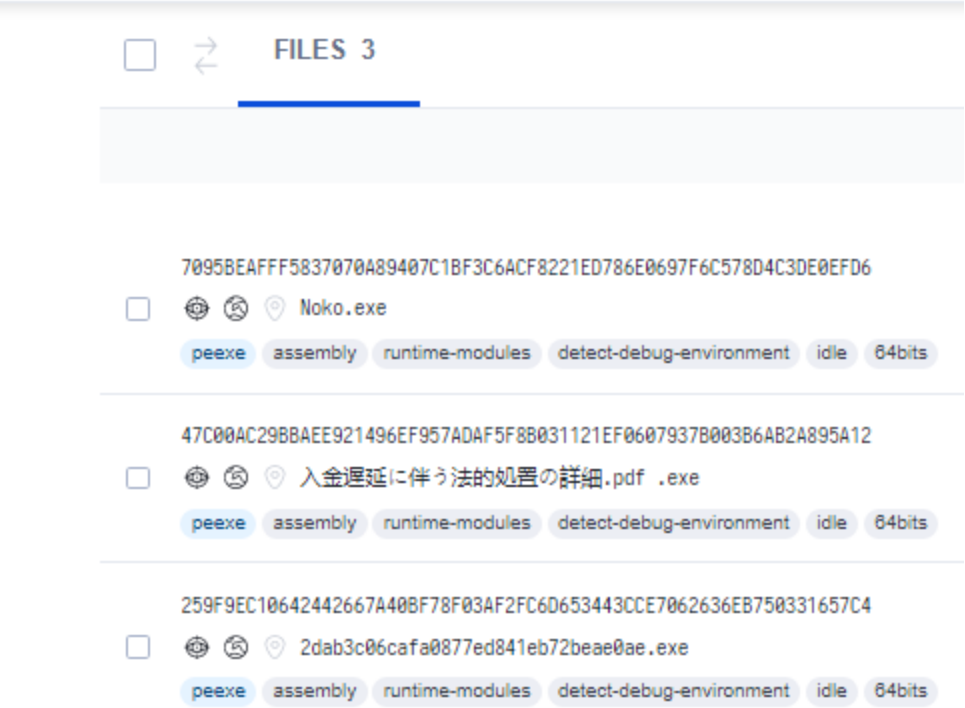
```
.rdata:000000... 0000001D     C     Error while formating path!\n
.rdata:000000... 0000000A     C     SKIP_EXTS
.rdata:000000... 00000013     C     DELETE_SHADOW\\\\.\\:
.rdata:000000... 00000028     C     Successfully deleted shadow copies from
.rdata:000000... 00000038     C     Your config isn't configurated to delete shadow copies!\n
.rdata:000000... 00000027     C      was found. Added to encryption list.\n
.rdata:000000... 0000004C     C     src\\windowsapi.rsCouldn't delete shadow copies from volume! GetLastError: \n
.rdata:000000... 00000064     C     Q:\\W:\\E:\\R:\\T:\\Y:\\U:\\I:\\O:\\P:\\A:\\S:\\D:\\F:\\G:\\H:\\J:\\K:\\L:\\Z:\\X:\\C:\\V:\\B:\\N:\\M:\\INVALID_HANDLE_VALUE\n
.rdata:000000... 00000050     C     /rustc/a55dd71d5fb0ec5a6a3a9e8c27b2127ba491ce52\\library\\core\\src\\str\\pattern.rs
.rdata:000000... 0000002C     C     called `Result::unwrap()` on an `Err` value
.rdata:000000... 0000000D     C     How to run:\n
.rdata:000000... 00000040     C        --config <base64 encoded config> (to start full encryption)\n
.rdata:000000... 0000000C     C     src\\main.rs
.rdata:000000... 0000004D     C      --config <base64 encoded config> --file <filePath> (encrypt selected file)\n
.rdata:000000... 00000050     C      --config <base64 encoded config> --dir <dirPath> (encrypt selected directory)\n
.rdata:000000... 00000024     C     CIS lang detected! Stop working...\n
.rdata:000000... 0000004A     C     ;LOAD_HIDDEN_DRIVESYour config isn't configurated to load hidden drives!\n
.rdata:000000... 0000004A     C     ENCRYPT_NETWORKYour config isn't configurated to encrypt network shares!\n
```

- “deps\noko.pdb”
- “How to run:”
- ”–config (to start full encryption)"
- ”–config --file "
- “CIS lang detected! Stop working”
- “config isn’t configurated to load hidden drives”
- “ENCRYPT_NETWORKYour config isn’t configurated to encrypt network shares”
- “Your config isn’t configurated to delete shadow copies”
- “Successfully deleted shadow copies”

By analyzing strings of malware, we can extract relevant strings and use VirusTotal (if you have a premium account) to test them individually in order to select appropriate conditions for our rules

We can detect that the file “deps\noko.pdb” will be present in all samples because it is a member of the family of pdb files.

```
content: "deps\\noko.pdb"
```



- PDB stands for “Program Database,” and it is a file format used by Microsoft Visual Studio to store debugging information about a program. It contains information about the program’s code, data, and resources, as well as details about the program’s execution. PDB files are typically used by developers to debug their programs and fix errors. They can also be used by other tools, such as debugger programs, to analyze the code and execution of a program. PDB files are often associated with the .exe file of a program, and they are typically stored in a separate directory or folder.
- We can use the PDB as a condition for detecting the presence of the Nokoyawa family in a sample. If the Yara scan identifies PDB in the sample, it will be identified as belonging to the Nokoyawa family.

After testing each string individually, we discovered that the first four strings were present in three samples, while the remaining strings were present in only one sample. Based on this information, we can conclude that the first three strings are except to the PDB string, and can therefore be used to detect the presence of the three samples. Therefore, our condition will be as follows: `uint16(0) == 0x5A4D and ($pdb or 3 of ($s*))`

Our YARA rule

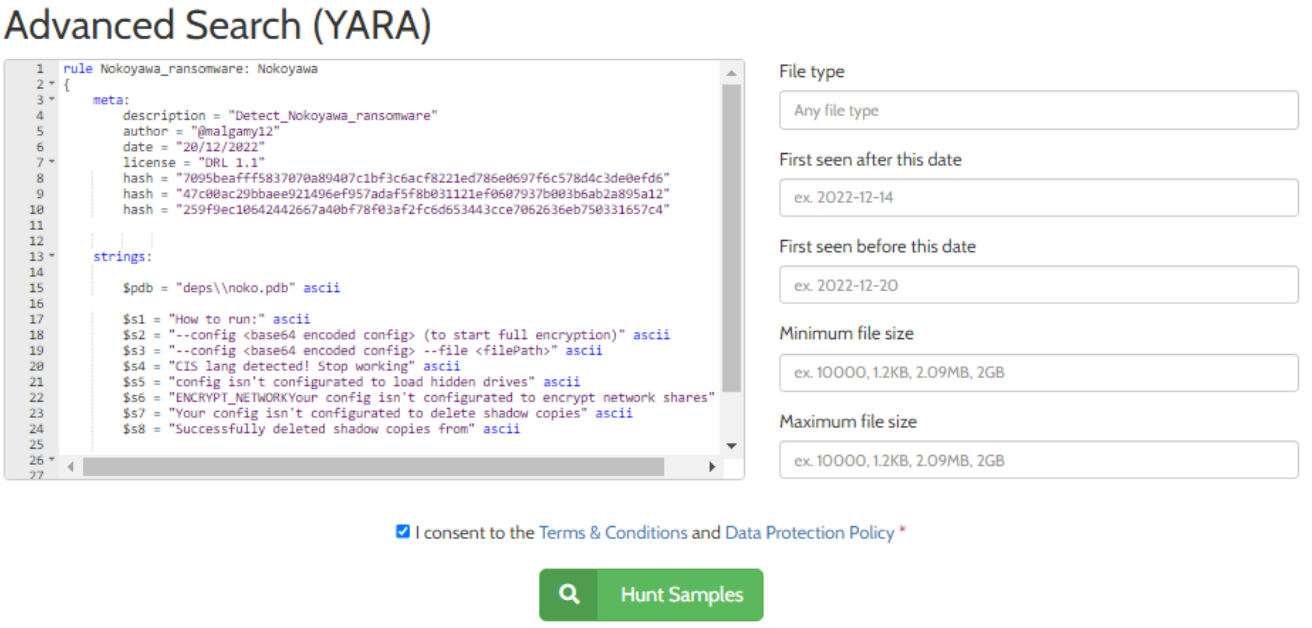
```
rule Nokoyawa_ransomware: Nokoyawa
{
    meta:
    description = "Detect_Nokoyawa_ransomware"
    author = "@malgamy12"
    date = "20/12/2022"
    license = "DRL 1.1"
        hash = "7095beafff5837070a89407c1bf3c6acf8221ed786e0697f6c578d4c3de0efd6"
        hash = "47c00ac29bbaee921496ef957adaf5f8b031121ef0607937b003b6ab2a895a12"
        hash = "259f9ec10642442667a40bf78f03af2fc6d653443cce7062636eb750331657c4"


    strings:

        $pdb = "deps\\noko.pdb" ascii
        $s1 = "How to run:" ascii
        $s2 = "--config <base64 encoded config> (to start full encryption)" ascii
        $s3 = "--config <base64 encoded config> --file <filePath>" ascii
        $s4 = "CIS lang detected! Stop working" ascii
        $s5 = "config isn't configurated to load hidden drives" ascii
        $s6 = "ENCRYPT_NETWORKYour config isn't configurated to encrypt network
shares" ascii
        $s7 = "Your config isn't configurated to delete shadow copies" ascii
        $s8 = "Successfully deleted shadow copies from" ascii

    condition:
        uint16(0) == 0x5A4D and ($pdb or 3 of ($s*))
}
```

Testing

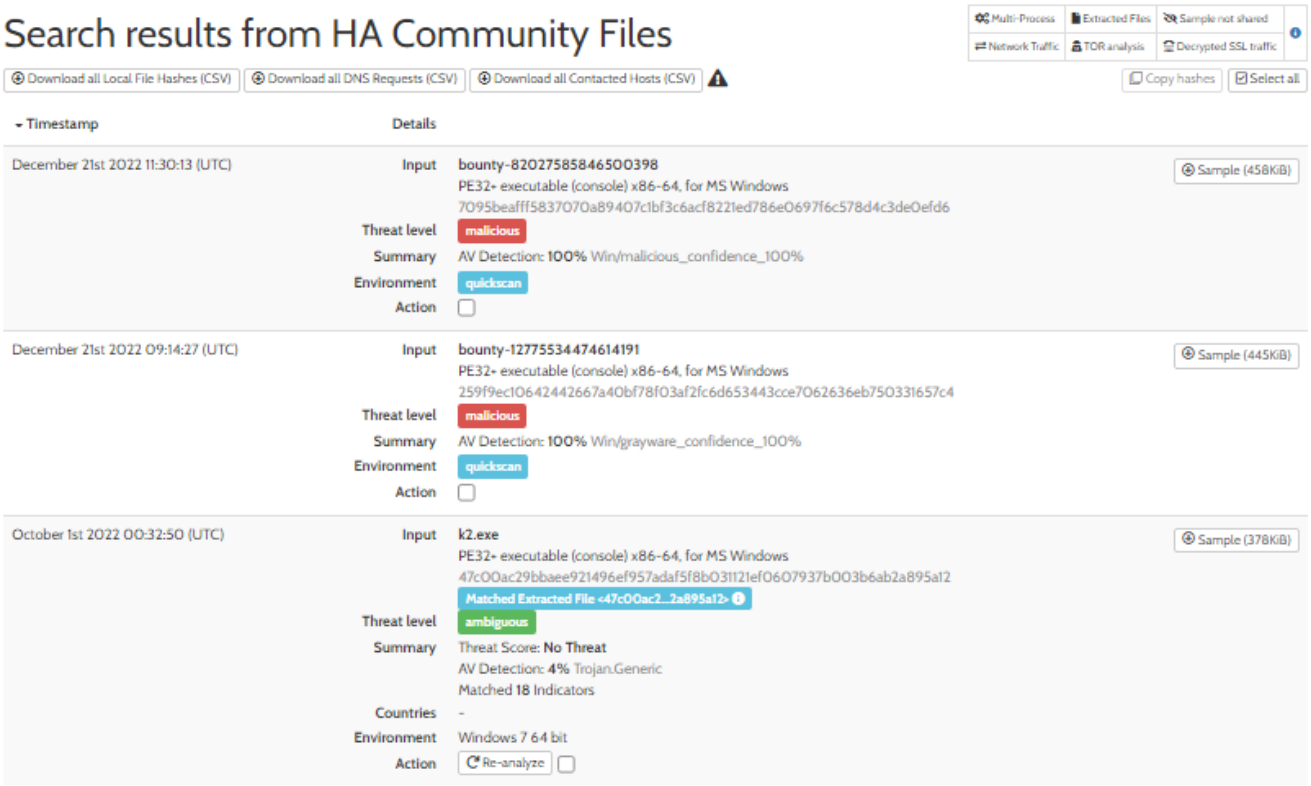
```
E:\yara\Nokoyawa>yara64.exe -r -s Nokoyawa.yara sample1
Nokoyawa_ransomware sample1
0x61372:$pdb: deps\noko.pdb
0x45de0:$s1: How to run:
0x45e04:$s2: --config <base64 encoded config> (to start full encryption)
0x45e89:$s3: --config <base64 encoded config> --file <filePath>

E:\yara\Nokoyawa>yara64.exe -r -s Nokoyawa.yara sample2
Nokoyawa_ransomware sample2
0x51cf2:$pdb: deps\noko.pdb
0x3e160:$s1: How to run:
0x3e184:$s2: --config <base64 encoded config> (to start full encryption)
0x3e209:$s3: --config <base64 encoded config> --file <filePath>

E:\yara\Nokoyawa>yara64.exe -r -s Nokoyawa.yara sample3
Nokoyawa_ransomware sample3
0x63fbb:$pdb: deps\noko.pdb
0x48490:$s1: How to run:
0x484b4:$s2: --config <base64 encoded config> (to start full encryption)
0x48539:$s3: --config <base64 encoded config> --file <filePath>
0x48630:$s4: CIS lang detected! Stop working
0x486b0:$s5: config isn't configurated to load hidden drives
0x486f8:$s6: ENCRYPT_NETWORKYour config isn't configurated to encrypt network shares
0x481d8:$s7: Your config isn't configurated to delete shadow copies
0x48190:$s8: Successfully deleted shadow copies from
```

As depicted in the preceding figure, it appears that our condition is functioning as intended. After conducting testing, we can confidently assert that our rules are effective on our sample set.

Hunting



From the previous figure, we can see the results of our rules



Thanks a lot for reading. You can find me into the following links